M. Briscoe

WHOI-77-75

Woods Hole Oceanographic Institution

WOODS HOLE OCEANOGRAPHIC INSTITUTION 1930

AUTO AND CROSS-BISPECTRAL ANALYSIS
OF SCALAR AND VECTOR TIME SERIES:
PROGRAMS, PROGRAM DESCRIPTIONS, AND TESTS
WITH ARTIFICIAL DATA

by

Gerard H. Martineau
and
Melbourne G. Briscoe

April 1978

TECHNICAL REPORT

Prepared for the National Science Foundation under Grant OCE76-14739.

WOODS HOLE, MASSACHUSETTS 02543

JGK

WHOI-77-75

AUTO AND CROSS-BISPECTRAL ANALYSIS OF SCALAR AND VECTOR TIME SERIES: PROGRAMS, PROGRAM DESCRIPTIONS, AND TESTS WITH ARTIFICIAL DATA

by

Gerard H. Martineau
and
Melbourne G. Briscoe

WOODS HOLE OCEANOGRAPHIC INSTITUTION
Woods Hole, Massachusetts 02543

Accesion For		
NTIS CRA&I		☒
DTIC TAB		☐
Unannounced		☐
Justification		
By		
Distribution /		
Availability Codes		
Dist	Avail and / or Special	
A-1		

April 1978

TECHNICAL REPORT

Prepared for the National Science Foundation under Grant OCE76-14739.

Approved for Distribution: Valentine Worthington
Valentine Worthington, Chairman
Department of Physical Oceanography

Table of Contents

I. Introduction, Summary, and Examples

A. Introduction

The usual spectral analysis is an examination of the second-order properties of a time series. The spectrum is, in fact, a description of how the various frequency bands contribute to the variance (the second moment) of the series, or the covariance in the case of cross-spectra. If the time series is a record from a Gaussian process, then the second-order spectrum provides a "complete" description because all higher moments of the series (for example the third moment, skewness, or the fourth moment, Kurtosis) are either zero or representable in terms of the second moment. One particular feature of stationary Gaussian processes is that the individual frequency components are independent; second-order spectral analysis is consistent with this feature, in that nothing in the analysis is sensitive to any non-independence of frequency components.

But phase-locking of frequency components is a significant feature of numerous important processes. For instance, spectral lines may show up at some frequency and at twice that frequency; second-order spectral analysis can only detect the existence of the lines, but an analysis that can detect whether the two lines are phase-locked can tell if the higher-frequency line is simply the first harmonic of the lower-frequency line. A common example would

be lines at 0.08 and 0.16 cph; the first is probably the M_2 tide, but is the second the M_4 tide or the first harmonic of the M_2 tide? A less common example would be quadratic nonlinear interactions, in which the product of frequency components at ω_1 and ω_2 would give rise to a phase-locked frequency component at ω_3, where $\omega_1 + \omega_2 - \omega_3 = 0$. This is analogous to the usual spectral analysis in which only two frequency components contribute to the energy, such that $\omega_1 + \omega_2 = 0$; this is usually written as $\omega_2 = -\omega_1$, and interpreted as the energy being due to the frequency components at $\pm\omega_1$.

Thus the bispectrum is a third-order spectrum (c.f., Hasselman, Munk, and McDonald, 1963;[1] Nagata, 1970;[2] Yao, Neshyba, and Crew, 1975[3] and 1977;[4] and Haubrich, 1965[5]) that is sensitive to phase-locked frequency triplets, such that $\omega_1 + \omega_2 - \omega_3 = 0$. Because only ω_1 and ω_2 need be mentioned ($\omega_3 = \omega_1 + \omega_2$ always), the pair (ω_1, ω_2) is referred to as a bi-frequency, and the display of the bispectrum is usually in terms of contours in the bi-frequency plane.

If all three frequencies from the triplet $(\omega_1, \omega_2, \omega_3)$ are from a single scalar (real) series, it is becoming customary to refer to the "auto-bispectrum," whereas if the three frequencies are not from the same scalar series, one refers to the "cross-bispectrum." If the series are not scalar, but rather vector random processes, such as

wind or ocean current velocity, one can choose a method of treating the vectors developed by Yao et al., 1975,[3] and use a rotary decomposition of the vectors that is independent of the rectangular coordinate system that might be used, and which is particularly useful for physical processes involving rotation. In this case one refers to "auto-rotary bispectra," and so on.

B. Summary

This report describes a series of computer programs developed at Woods Hole Oceanographic Institution during 1977 to compute scalar bispectra and cross-bispectra (BISCAL), rotary bispectra and cross-rotary bispectra (BIVEC) and to display corresponding bicoherences (BPLOT) and rotary bicoherences (RBPLOT) as contour plots on the bi-frequency plane.* The usual method of calculating the bispectrum in the frequency domain after using the Fast Fourier Transform is used here, and so there is a program for calculating the Fourier coefficients (FOURIER) as well as programs for organizing the data (FRAGTAP, ORDAT) or generating artificial series (GENRAN) if desired. There is also one special-purpose program which calculates line integrals of (scalar) bicoherences in the bifrequency plane along paths of constant $\omega_3 = \pm\omega_1 \pm\omega_2$ to determine the relative total contribution of quadratic interactions to each frequency (BISUM). After the descriptions of these programs,

*See pages 75-77 for precise definitions of the bispectrum and bicoherence.

there is a collection of actual FORTRAN-IV listings of them arranged alphabetically.

C. Examples Based on Artificial Signals

Definition of "Hanning" and "Overlapping"

In the following discussion, the terms "Hanning" and "overlapping" are used. Hanning* refers to a particular data window applied to the time domain in order to reduce side lobes and consequent leakage in the frequency domain. The window is a "raised cosine" over 100 percent of the time record; many windows are possible and some are more efficacious, but the Hanning window is extremely efficient to implement. See Reference 6 for a general discussion of windows.

One effect of windowing is to lessen the effective length of the time domain segment, which equivalently reduces the useful degrees of freedom in the segment. This effect is compensable by segmenting the entire time domain record into overlapping pieces so that the losses due to windowing in one segment are recovered in the overlapped segment. Nuttall[7] has shown that 62.5 percent overlapping allows a complete recovery of degrees-of-freedom losses, but that 50 percent overlapping is much simpler and yields a 92 percent recovery of losses.

The computational recipes applied here are as follows:

---

*"Hanning" is a traditional misnomer: the term is based on the name of Julius Von Hann, who first described the window, by analogy to Hamming, which is another window that is due to Richard Hamming.

Let a series consist of $(2T_m)$ points spaced at a "time" $\Delta\tau = 1$ apart so that the Nyquist frequency is (1/2) and there are (T_m+1) spectral estimates. Let $u'_{T_m}(\tau)$ be the Hanning window to be applied at time τ. Then:

$$u'_{T_m}(\tau) = \begin{cases} 0 & \tau < 0 \\ \frac{1}{2}\left[1-\cos\left(\frac{\pi\tau}{T_m}\right)\right] & 0 \le \tau \le 2\,T_m \\ 0 & \tau > 2\,T_m \end{cases} \qquad (1)$$

The Fourier transform of $u'_{T_m}(\tau)$ is $U'_{T_m}(f)$ as follows:

$$U'_{T_m}(f) = \frac{1}{2}U_{T_m}(f) - \frac{1}{4}U_{T_m}\left(f - \frac{1}{2T_m}\right) - \frac{1}{4}U_{T_m}\left(f + \frac{1}{2T_m}\right) \qquad (2)$$

where $0 \le f \le 1/2$ and $\Delta f = 1/(2T_m)$ and $U_{T_m}(f)$ is the Fourier transform of the rectangular window of unit height over the time $2T_m$ for which the expression (1) is nonzero.
By the convolution theorem the application of expression (2) to the Fourier transformed series is equivalent to applying (1) to the original series. It is expression (2) that is used to accomplish Hanning below. Specifically, if we consider a real series of n points having the following Fourier representation:

$$U_i = a_o + \sum_{k=1}^{n/2} [a_k \cos(2\pi jk/n) + b_k \sin(2\pi jk/n)],$$

then the i^{th} Hanned coefficient is:

$$U'_i = 0.5\,U_i - 0.25(U_{i-1} + U_{i+1})\ .$$

The Fourier coefficients a_1 through $(a_{n/2} - 1)$, and b_2 through $b_{n/2 - 2}$ are changed by Hanning, leaving the following unchanged: a_o, $a_{n/2}$, b_1, aNd $b_{n/2-1}$.

If the variance of the Hanned series is now computed, it will be found to be low because of the cosine taper. To correct for this, the transformed and Hanned series can be multiplied by the ratio of the variance of the transformed, non-Hanned, series to the variance of the transformed, Hanned series. An approximation to this value, valid for a sufficiently large number of data pieces, is a theoretical value[2] for Gaussian, white noise: $\sqrt{8/3}$. All transformed and Hanned series in this report have been multiplied by $\sqrt{8/3}$.

In calculating the power spectrum of a data series, it is desirable to divide the series into segments, and to overlap these segments when using a data or spectral window such as the Hanning window (Welch, 1967[9]). The segmenting is desirable for many reasons, such as the reduction in the series length to be transformed, a possible reduction in the number of overall computations, greater ease of observing nonstationary trends in the data, and greater control over frequency resolution.

Specifically, for a 50% overlap, let the number of points per segment or "piece" be S. If the pieces are numbered consecutively i = 1,2,..., and the data points are numbered consecutively 1,2,..., then the i^{th} piece starts

at point [1+(i-1)S/2]. This is done until no more complete pieces can be constructed from the data, the remaining data being discarded if necessary. Note that because of the Fast Fourier Transform algorithm used, the piece length is never odd.

All real (i.e., scalar) series or real components of vector series to be used below consist of 10 pieces of length 64, resulting in 32 complex Fourier coefficients for each piece. These 10 pieces are contiguous in the case of no Hanning and no overlapping, meaning that 640 points of the series are used. For Hanning and 50% overlapping, 352 points of the series are used.

All contour plots except Figures 16 and 17 are contoured at the 80, 85, 90, 95, and 99% confidence levels determined from a Gaussian noise background.* Figures 16 and 17 emphasize the bicoherence peaks by displaying contours at the 95, 96, 97, 98, and 99% confidence levels. All series used for auto- and cross-bicoherence plots are Hanned and overlapped. A comparison of Hanning with no Hanning is given in the demonstration of auto-rotary bicoherence plots.

---

*See page 7 in program BISCAL report below, and pages 6,7 in program BIVEC below.

1. Auto-bicoherence

Figures 3, 4, and 5 are auto-bicoherence contour plots of the following three Hanned artificial signals:*

Fig. 3: $s = g + a(\cos 8t + \cos 16t + \cos 24t)$

Fig. 4: $s = g + a(\sin 8t + \sin 16t + \sin 24t)$

Fig. 5: $s = g + a(-\sin 8t - \sin 16t - \sin 24t)$

In all three cases, g is Gaussian noise with zero mean and unity variance, and a = 10. Figure 3 uses the Gaussian noise background whose auto-bicoherence is displayed in Figure 1, while Figures 4 and 5 both use the Gaussian noise background having the auto-bicoherence shown in Figure 2. It is observed that all three figures are essentially alike, with two peaks at the only possible coupled frequency triplets $(\omega_1, \omega_2, \omega_3)$:

$$(\omega_1, \omega_2, \omega_3) = (8, 8, 16) \text{ and } (8, 16, 24) .$$

Note that ω_3 is the horizontal intercept of the straight line $\omega_2 = -\omega_1 + \omega_3$ [i.e., $\omega_1 + \omega_2 - \omega_3 = 0$] having slope (-1) and passing through the bi-frequency (ω_1, ω_2). The contour plots are alike because the signals differ only in the phasing of the terms and the bicoherence does not give any phase information.

*In generating all of the artificial series used in this report, an error in indexing in program GENRAN introduced into each sinusoidal term an additional phase angle of $(-2\pi\omega/L)$, where ω = frequency and L = piece length. Since this made no difference in the appearance or interpretation of the plots, the examples presented were not repeated.

Now notice that if Figures 3 and 4 were presented as the "east" and "north" components of velocity, for example, they would be representing three coupled counterclockwise rotating vectors of frequencies 8, 16, and 24. If Figures 3 and 5 were so presented, they would be representing three coupled clockwise rotating vectors. Since the pairs of figures are identical, could one detect the presence of these coupled rotating vectors? In order to do this, it is necessary to examine the biphase at the coincident peaks in the east and north components. The bicoherence and rotary bicoherence programs can furnish this information. For Figure 3 the biphase value for all peaks is essentially 0. For all peaks of Figure 4, it is $\pi/2$, and for all peaks of Figure 5, it is $-\pi/2$. This means that for the coincident peaks in the pair consisting of Figures 3 and 4, the north biphase leads east by 90° and for the pair 3 and 5, north lags east by 90°. The fact that there are coincident peaks implies the existence of coupled sinusoids of the same frequency. The biphase difference of 90° implies that the sinusoids can represent the east and north components of a rotating vector, and the sign of the difference gives the sense of rotation. Thus the conditions above can exist together only if Figures 3 and 4, given as east and north, represent a coupled counterclockwise-rotating vector, and if Figures 3 and 5 represent a coupled clockwise-rotating vector.

It will be demonstrated below that the same conclusions about coupled rotating vectors and their sense of rotation can be reached by examining an auto-rotary bicoherence plot of these pairs of signals.

Figure 6 is an auto-bicoherence plot of the Hanned, overlapped, artificial signal

$$s = g^2 + 10\ g\cdot\sin(16t) - 50\ \cos(32t)$$

where g is Gaussian noise of zero mean and unity variance. The ridge pattern in Figure 6 comes from the joint behavior of the second and third terms in the signal. The second term is a product in the time domain of g (white noise, i.e., contains all frequency components) and a sinusoid (a delta function at 16), which could be expressed as a convolution in the frequency domain. The convolution reproduces the delta function at <u>all</u> the frequencies, not just 16. The sin(16t) and the cos(32t) are phase-locked, because no randomness is introduced into the phases, hence the g·sin(16t) is also phase-locked to the third term.

Noting that $\omega_1 + \omega_2 - \omega_3 = 0$ is a requirement for a significant bispectrum, and noting that the second term describes a signal that may be thought of as a modulation of a signal at 16 to the new frequencies $16 + \omega_i$ and $16 - \omega_i$ and letting $\omega_1 = 16 + \omega_i$, $\omega_2 = 16 - \omega_i$, and $\omega_3 = 32$, we see that $\omega_1 + \omega_2 - \omega_3 = 0$ for <u>all</u> ω_i! Also,

$\omega_1 + \omega_2 - \omega_3 = 0$ if $\omega_i = 16$. Therefore, we obtain significant bicoherence whenever $\omega_1 + \omega_2 = 32$ <u>or</u> when either $\omega_1 = 32$ or $\omega_2 = 32$.

2. Cross-bicoherence

Figures 7 and 7a are cross-bicoherence plots of the following two Hanned signals, s_x and s_y:

$$s_x = g_x + a(\cos 8t + \cos 16t + \cos 24t)$$

$$s_y = g_y + a(\sin(-8t) + \sin(-16t) + \sin(-24t)) \ .$$

Here g_x is the Gaussian noise of Figure 1 and g_y is the Gaussian noise of Figure 2. The cross-bicoherence between g_x and g_y is shown in Figure 7b. The form is $g_x g_x g_y$ for $\omega_1 \omega_2 \omega_3$ (see below). The amplitude <u>a</u> is again 10. Figure 7a displays bicoherences normalized by amplitude as described in the report on program BISCAL. Notice first that the domain has increased because the portion of the bi-frequency plane in which the bicoherence is unique has increased for cross-bicoherence.* The cross-bicoherence is computed here by using s_x as the source of ω_1, s_x again as the source for ω_2, and s_y for ω_3. The source signals could have been taken in the only other unique sequence $s_x s_y s_y$. It can be seen that there are peaks at the only possible coupled frequency triplets:

---

*See page 5 of the program BISCAL report below.

$$(\omega_1, \omega_2, \omega_3) = (8,8,16),(8,16,24),(-8,16,8),$$
$$(-8,24,16), \text{ and } (-16,24,8) \ .$$

Notice that in effect s_x and s_y contain both positive and negative frequencies; i.e., 8, 16, 24, -8, -16, -24. There are "edges" of peaks at the frequency triplets (-8,32,24), (-16,32,16), and (-24,32,8). These are visible because of the frequency-spreading due to Hanning, and are in the portion of the bi-frequency plane which is redundant with the domain plotted.

Figure 7c is a plot of average bicoherence where the average is over N, the number of points used in calculating a line integral over the diagonal straight line $\omega_2 = -\omega_1 + \omega_3$ corresponding to the locus of all frequency triplets having a given "sum frequency" ω_3 . The abscissa is the sum frequency and the plot is directly based on Figure 7. The integrals are calculated by program BISUM. It is seen that there are peaks at 16 and 24 for positive ω_2, corresponding to the two frequency triplets (8,8,16) and (8,16,24). The apparent peak at $\omega_3 = 2$ is due to the fact that only one point contributes to this sum, hence the "average" is poorly specified. For negative ω_2, there is a peak at 8 corresponding to the frequency triplets (-8,16,8) and (-16,24,8) and another at 16 corresponding to (-8,24,16). The peak at 24 is due solely to the redundant bicoherence peak at the frequency triplet (-8,32,24) as mentioned above.

Figure 7d is a plot of average squared-bicoherence based on Figure 7, which emphasizes the peaks.

3. Auto-rotary Bicoherence

Figures 8 and 9 are auto-rotary bicoherence plots of Gaussian noise signals used as background noise in the plots having contours at the confidence levels 80, 85, 90, 95, 99% (all except Figure 17), and for determining these confidence levels. These Gaussian noise signals are processed in the same way as the signals in which they serve as noise. Figure 8 differs from Figure 9 in that the noise of Figure 8 has been Hanned and overlapped, while that of Figure 9 has undergone neither. It is observed that both plots appear to have a random distribution of peaks.

Figures 10, 11, 12, and 13 are auto-rotary bicoherence plots of the following complex signal: $\vec{s} = (g_1+au) + i(g_2+av)$

$$u = \cos 8t + \cos 16t + \cos 24t + \cos(-8t) + \cos(-16t) + \cos(-24t)$$
$$= 2(\cos 8t + \cos 16t + \cos 24t)$$

$$v = \sin 8t + \sin 16t + \sin 24t + \sin(-8t) + \sin(-16t) + \sin(-24t)$$
$$= 0$$

g_1, g_2 = Gaussian noise, mean = 0, variance = 1

a = 10 for Fig. 10, 1.0 for Fig. 11, 0.1 for Fig. 12, 10 for Fig. 13.

This complex signal has three counter-rotating vectors of frequencies 8, 16, 24. In Figures 10, 11, and 12 the signal

has been Hanned and the signal-to-noise ratio (S/N) is 10, 1.0 and 0.1 respectively. There are marked peaks at the same frequencies in Figures 10 and 11, but Figure 12, where S/N is down to 0.1, does not seem to differ significantly from Gaussian noise. It is apparent from the relative sizes of the contours that they are higher on the peaks of Figure 11 (S/N = 1.0) than on those of Figure 10 (S/N = 10) as expected. There are peaks at the following coupled frequency triplets:

(8,8,16), (8,16,24)	quadrant I
(16,-8,8), (24,-8,16), (24,-16,8)	quadrant II
(-16,-8,-24), (-8,-8,-16)	quadrant III
(-16,8,-8), (-24,8,-16), (-24,16,-8)	quadrant IV

In a rotary bicoherence plot, a peak indicates a coupling of two rotating vectors of frequencies ω_1 and ω_2 with a third of frequency ω_3. The sign of the frequency gives the sense of rotation in all cases, positive indicating counterclockwise. Figures 10 and 11 demonstrate that there are indeed three counter-rotating vectors of frequencies 8, 16, 24, -8, -16, -24 which are pairwise coupled so that the sum and difference frequencies exist in the same set.

Figure 13 is an auto-rotary bicoherence plot of the same signal with a S/N = 10, but this time not Hanned. Comparing it with the Hanned case of Figure 10, it is apparent that the peaks are still present at the same frequencies, but they are much narrower because they are each due to a

single point. The peaks (8,8,16) and (-8,-8,-16) are ill-formed because the one contributing point lies on the boundary of the contour plot. In general, these peaks are difficult to distinguish from the Gaussian noise background, and they would require something like a stereoscopic or perspective display. Alternatively one can raise the contour levels to emphasize the peaks and suppress the noise background. This is done in Figure 17 in which the contour levels are at the 95, 96, 97, 98, and 99% confidence levels, and the highlighting is apparent. The noise background itself is separately displayed in Figure 16.

Figures 14 and 15 are for comparison with Figures 3, 4, and 5 in connection with a hypothetical presentation of Figures 3 and 4, and 3 and 5 as two "east" and "north" component pairs of auto-bicoherence plots. Figure 14 is an auto-rotary bicoherence plot using a vector series whose u and v components are identically the signals of Figures 3 and 4 respectively. Figure 14 displays two peaks, at (8,8,16) and (16,8,24). These indicate directly that counterclockwise pairs of rotating vectors of rotation frequencies (8,8) and (16,8) are phase-locked to counterclockwise rotating vectors of frequencies 16 and 24 respectively. Recall that Figures 3 and 4 had to be used in conjunction with biphase information to arrive at the same conclusion. Figure 15 displays two peaks, at (-8,-8,-16) and (-16,-8,-24),

indicating a similar set of clockwise rotating vectors, for which conclusion biphase information had to be added to the auto-bicoherence plots of Figures 3 and 5.

Figure 18 is a cross-rotary bicoherence plot between the following two complex signals, called X and Y:

X and Y are of form $\vec{s} = (g_1+au) + i(g_2+av)$, where:

g_1, g_2 = Gaussian noise, mean = 0, variance = 1

$a = 10$

For X:

$u = \cos(-8) + \cos(-16) + \cos(16)$

$v = \sin(-8) + \sin(-16) + \sin(16)$

For Y:

$u = \cos 8 + \cos 16 + \cos(-24)$

$v = \sin 8 + \sin 16 + \sin(-24)$.

The cross-rotary bispectrum and bicoherence are computed with X supplying ω_1, X supplying ω_2, and Y supplying ω_3 ("Form XXY"). Figure 18 displays only two peaks, at the coupled frequency triplets (16,-8,8) and (-16,-8,-24). The first indicates phase-locking of a counterclockwise rotating vector of frequency 16 with a clockwise rotating

one of frequency (-8) to produce a counterclockwise-rotating vector of frequency 8. The second indicates coupling of three clockwise-rotating vectors, Examination of the input frequencies shows that these are indeed the only two possibilities.

D. Empirical Confidence Levels for Bicoherence

Figure 19 displays empirically determined confidence levels at 90, 95, 99, and 99.9 percent confidence that the bicoherence lies below the value given by the graph as a function of the number of pieces over which the averaging was done and the equivalent number of independent pieces. The data were supplied by program BISCAL, which produced a record of 4096 auto-bicoherence values from Gaussian noise having mean 0 and unity variance. The record was processed by splitting it into 256-point piece lengths, Hanning, and 50% overlapping. Program BISCAL placed the bicoherences into bins of width 0.01, determined the fraction of the total number (4096 here) lying below the upper bin limit, and calculated the above confidence levels by linearly interpolating between bins.

The dashed line in Figure 19 is the modified theoretical 95% confidence level determined from the following expression (Haubrich, 1965):[5]

$$95\% \text{ confidence level} = (6/2P')^{1/2}$$

where:

$2P'$ = equivalent degrees of freedom

P' = equivalent number of independent pieces

$= 18P^2/(19P-1)$

P = number of Hanned, 50% overlapped pieces

Examination of Figure 19 shows that 5.9 would be slightly better than 6.0 as the constant in Haubrich's expression.

The horizontal axis of Figure 19 is given both in P' and P. If one had neither Hanned nor overlapped,* then the P' scale should be used to determine confidence limits; if the pieces were Hanned and overlapped, the P scale is applicable.

The curves in Figure 19 are based on the usual version of bicoherence (see BISCAL write-up) which normalizes the modulus of the averaged bispectrum by the autospectral energy in each of the three frequency bands being examined. An alternate bicoherence calculation (see also BISCAL write-up) normalizes each bispectrum before the average is

---

*One should only overlap if windowing is also being done, otherwise the equivalent number of degrees of freedom may actually be reduced (Nuttall, personal communication, 1973).

calculated and the modulus taken; this is done to emphasize phase relations and minimize amplitude relations amongst the three frequency bands being examined. The confidence limits (for P = 9 and 39) are shown as solid symbols in Figure 19; for 95% confidence levels on "phase bicoherence" the usual 95% confidence level is satisfactory. One might also use Haubrich's simple expression with 5.8 as the constant.

E. References for Section I

1. "Bispectra of Ocean Waves," by K. Hasselman, W. H. Munk, and G. MacDonald, Time Series Analysis, edited by M. Rosenblatt, pp. 125-139, John Wiley & Sons, New York, 1963.

2. "Lag Joint Probability, Higher Order Covariance Function, and Higher Order Spectrum," by Yutaka Nagata, La mer (Bulletin de la Société franco-japonaise d'océanographie), 8(2), pp. 78-94, May, 1970.

3. "Rotary Cross-Bispectra and Energy Transfer Functions between Non-Gaussian Vector Processes I. Development and Example," by Neng-chun Yao, Steve Neshyba, and Henry Crew, Journal of Physical Oceanography, 5, 164-172, January, 1975.

4. "Rotary Cross-Bispectra and Energy Transfer Functions between Non-Gaussian Vector Processes II. Winds and Currents off the Oregon Coast," by Neng-chun Yao, S. Neshyba, and H. Crew, Journal of Physical Oceanography, 7, 892-903, November, 1977.

5. "Earth Noise, 5 to 500 Millicycles per Second," by Richard H. Haubrich, Journal of Geophysical Research, 70, 1415-1427, 1965.

6. "On the Use of Windows for Harmonic Analysis with the Discrete Fourier Transform," by Fredric J. Harris, Proceedings of the IEEE, 66(1), 51-83, January, 1978.

7. "Spectral Estimation by Means of Overlapped Fast Fourier Transform Processing of Windowed Data," by A. H. Nuttall, Naval Underwater Systems Center Report No. 4169, October 13, 1971.

8. "Energy Loss in Surface Wave Spectra Due to Data Windowing," by Melbourne G. Briscoe, Deep-Sea Research, 19, 665-667, 1972.

9. "The Use of Fast Fourier Transform for the Estimation of Power Spectra: A Method Based on Time Averaging over Short, Modified Periodograms," P. D. Welch, IEEE Transactions on Audio and Electroacoustics, AU-15, 70-73, 1967.

AUTO BICOHERENCE

ORIGINAL FILE NAMES:		VARIABLE NUMBERS:
FREQUENCY 1:	GENRANOUTPUT	0
FREQUENCY 2:	GENRANOUTPUT	0
FREQUENCY 3:	GENRANOUTPUT	0

PROCESSING HISTORY

NO LF'S 8*NO WDS DATA 352*PREW 0*SUBSAMP 0*NSUBSAMP 0*PC SIZE 64
*NO PCS 10*OLAP 1*ISUBT 0*HANN 1 *CREATED 19:20 DEC 21,'77

TIME OF PLOT:
19:41 DEC 21,'77

CONTOUR LEVELS AND PERCENT CONFIDENCE

0.430	80.0
0.458	85.0
0.479	90.0
0.552	95.0
0.667	99.0

GAUSS NOISE HANNED
6 SEED=3333

FILE BVPL

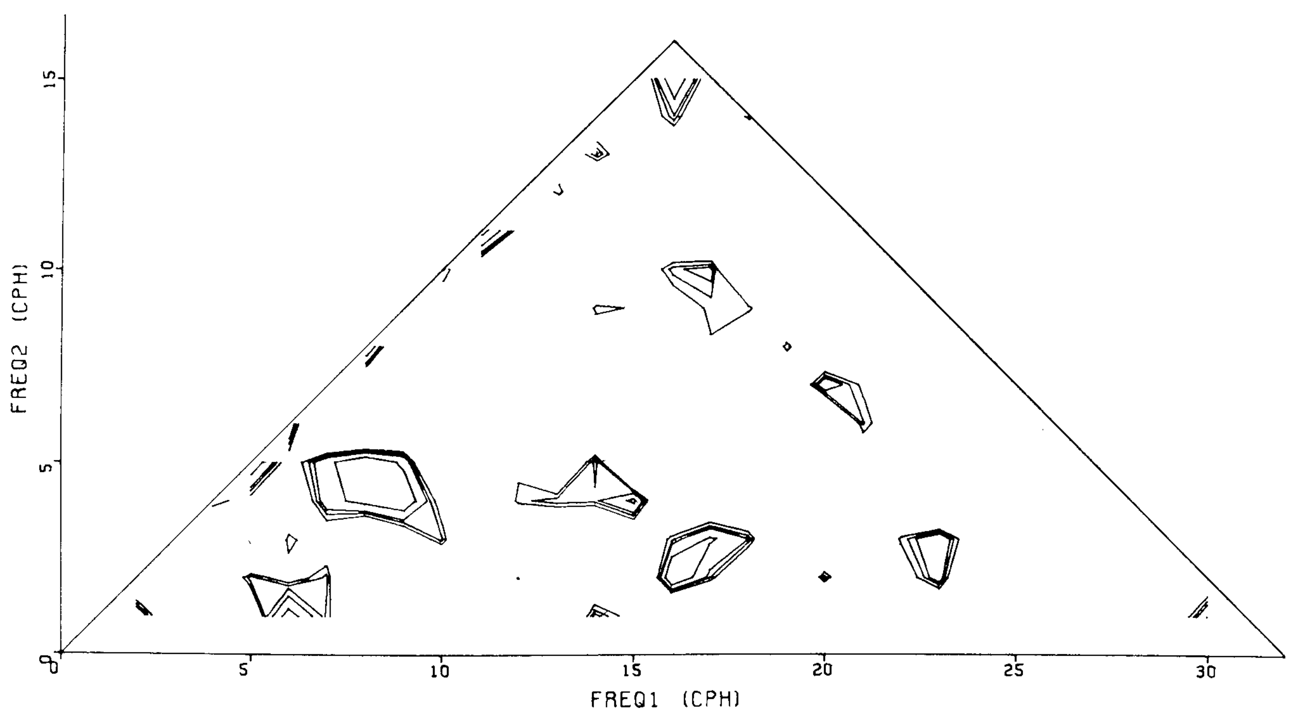


Figure 1

AUTO BICOHERENCE

ORIGINAL FILE NAMES:		VARIABLE NUMBERS:
FREQUENCY 1:	GENRANOUTPUT	0
FREQUENCY 2:	GENRANOUTPUT	0
FREQUENCY 3:	GENRANOUTPUT	0

PROCESSING HISTORY

NO LF'S 4*NO WDS DATA 352*PREW 0*SUBSAMP 0*NSUBSAMP 0*PC SIZE 64 *NO PCS 10*OLAP 1*ISUBT 0*HANN 1 *CREATED 17:18 DEC 15,'77

CONTOUR LEVELS AND PERCENT CONFIDENCE

TIME OF PLOT:
09:48 DEC 27,'77

0.420	80.0
0.447	85.0
0.498	90.0
0.591	95.0
0.834	99.0

GAUSSIAN NOISE HANNED FILE BVPL5
SEED=7777

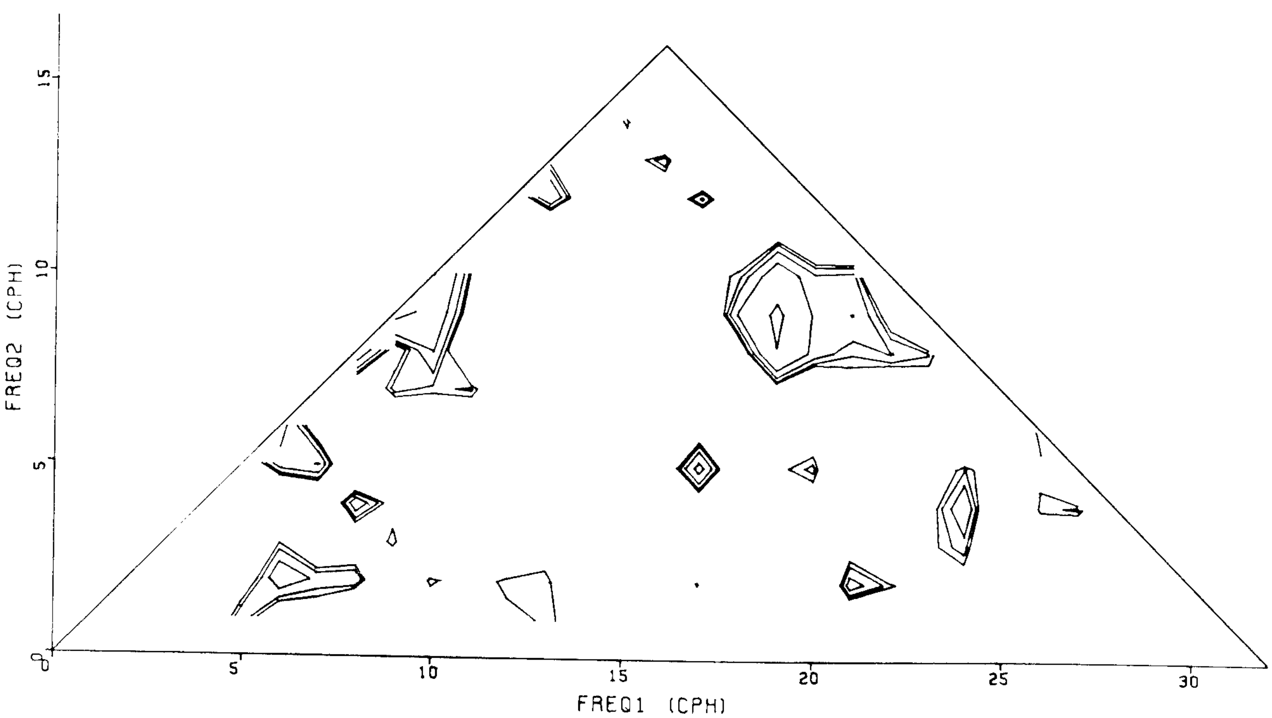


Figure 2

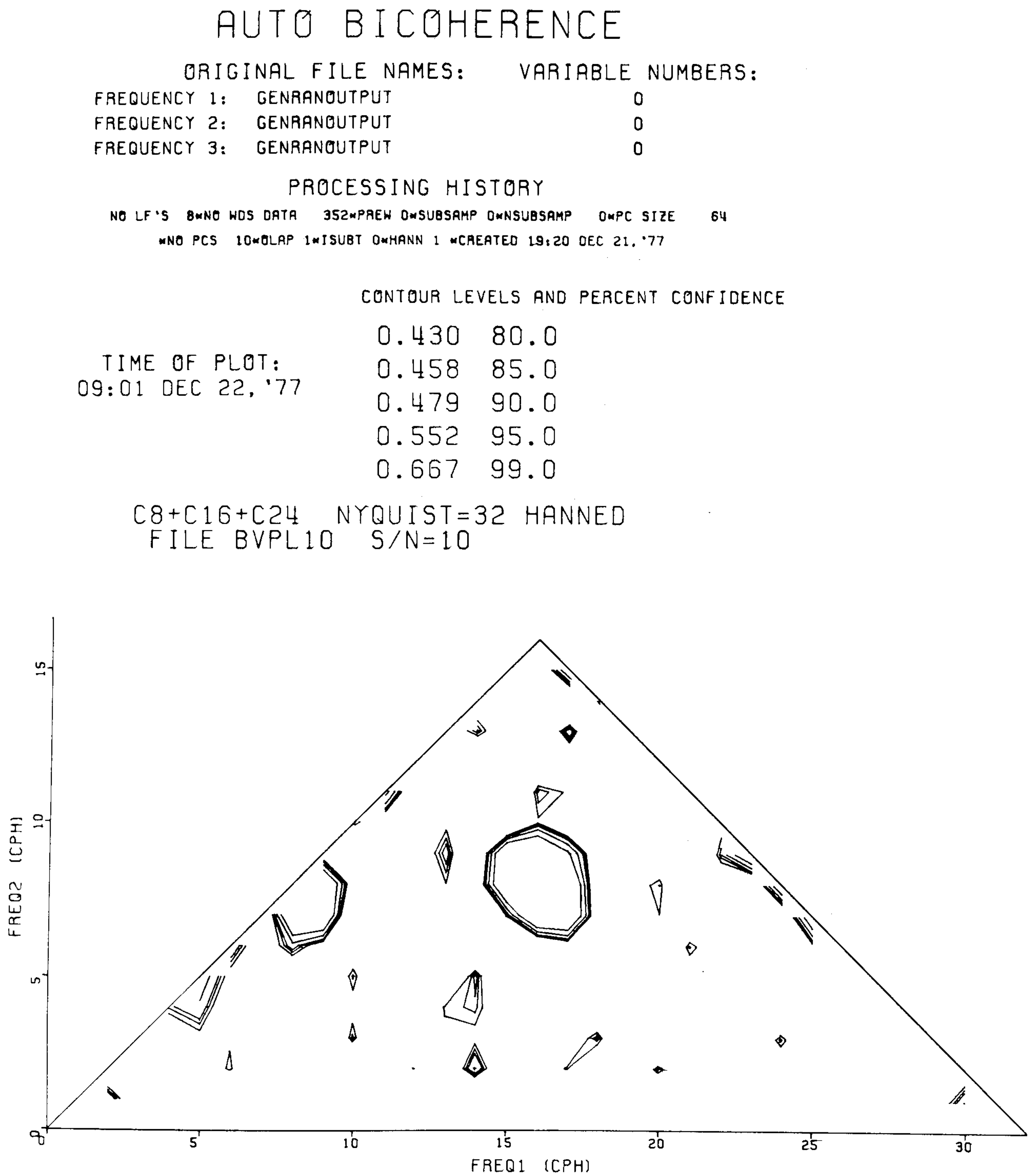

AUTO BICOHERENCE
ORIGINAL FILE NAMES: VARIABLE NUMBERS:
FREQUENCY 1: GENRANOUTPUT 0
FREQUENCY 2: GENRANOUTPUT 0
FREQUENCY 3: GENRANOUTPUT 0
PROCESSING HISTORY
NO LF'S 8*NO WDS DATA 352*PREW 0*SUBSAMP 0*NSUBSAMP 0*PC SIZE 64
*NO PCS 10*OLAP 1*ISUBT 0*HANN 1 *CREATED 19:20 DEC 21, '77
CONTOUR LEVELS AND PERCENT CONFIDENCE
0.430 80.0
0.458 85.0
0.479 90.0
0.552 95.0
0.667 99.0
TIME OF PLOT:
09:01 DEC 22, '77
C8+C16+C24 NYQUIST=32 HANNED
FILE BVPL10 S/N=10
FREQ2 (CPH)
FREQ1 (CPH)
0
5
10
15
20
25
30

Figure 3

AUTO BICOHERENCE

ORIGINAL FILE NAMES: VARIABLE NUMBERS:

FREQUENCY 1:	GENRANOUTPUT	0
FREQUENCY 2:	GENRANOUTPUT	0
FREQUENCY 3:	GENRANOUTPUT	0

PROCESSING HISTORY

NO LF'S 8*NO WDS DATA 352*PREW 0*SUBSAMP 0*NSUBSAMP 0*PC SIZE 64
*NO PCS 10*OLAP 1*ISUBT 0*HANN 1 *CREATED 19:20 DEC 21,'77

TIME OF PLOT:
09:06 DEC 22,'77

CONTOUR LEVELS AND PERCENT CONFIDENCE

0.420	80.0
0.447	85.0
0.498	90.0
0.591	95.0
0.834	99.0

S8+S16+S24 NYQUIST=32 HANNED
FILE BVPL11 S/N=10

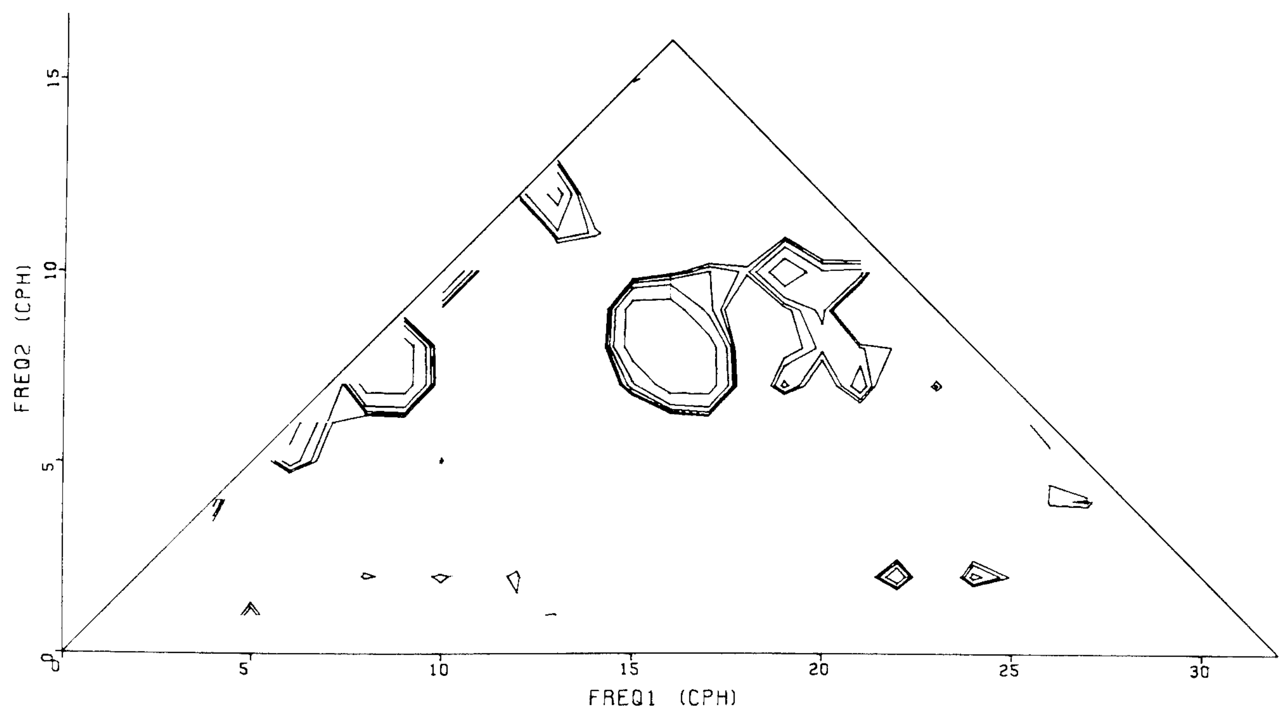


Figure 4

AUTO BICOHERENCE

ORIGINAL FILE NAMES: VARIABLE NUMBERS:

FREQUENCY 1:	GENRANOUTPUT	0
FREQUENCY 2:	GENRANOUTPUT	0
FREQUENCY 3:	GENRANOUTPUT	0

PROCESSING HISTORY

NO LF'S 8*NO WDS DATA 352*PREW 0*SUBSAMP 0*NSUBSAMP 0*PC SIZE 64
*NO PCS 10*OLAP 1*ISUBT 0*HANN 1 *CREATED 19:20 DEC 21, '77

TIME OF PLOT:
09:11 DEC 22, '77

CONTOUR LEVELS AND PERCENT CONFIDENCE

0.420	80.0
0.447	85.0
0.498	90.0
0.591	95.0
0.834	99.0

-S8-S16-S24 NYQUIST=32 HANNED
FILE BVPL12 S/N=10

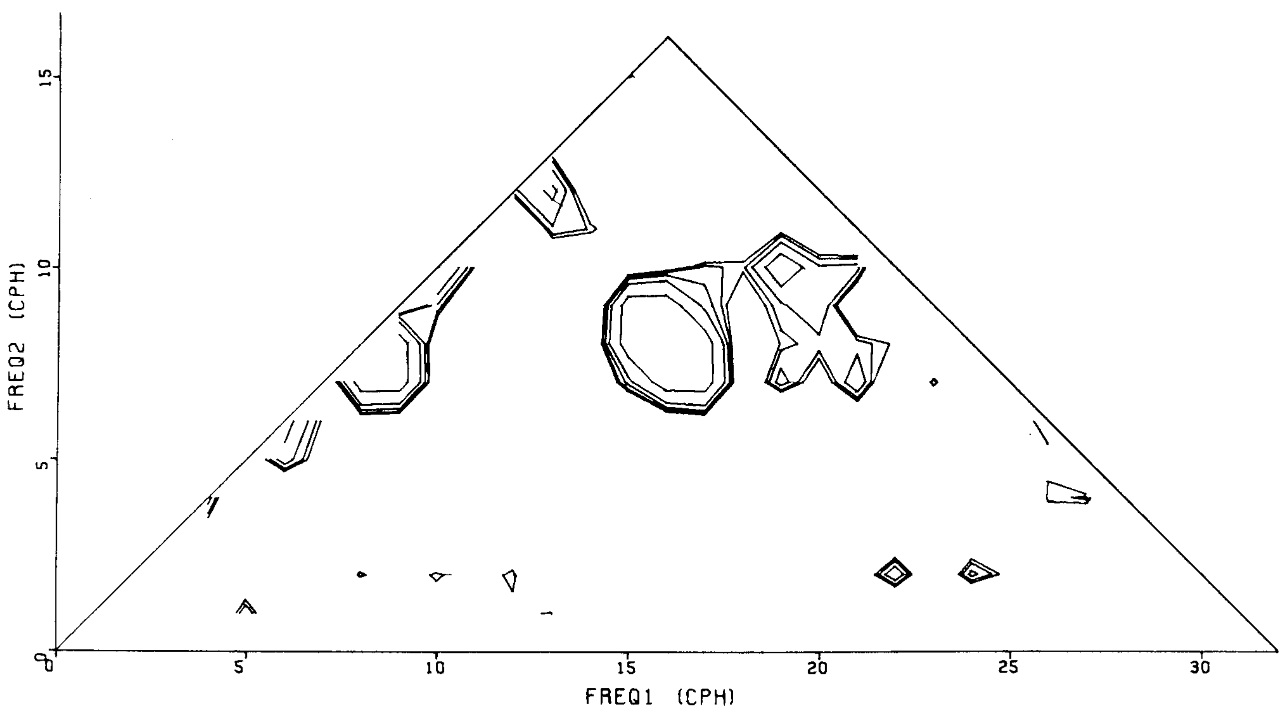


Figure 5

AUTO BICOHERENCE

CONTOUR LEVELS AND PERCENT CONFIDENCE

TIME OF PLOT:
15:58 MAR 07,'78

0.398	95.0
0.489	99.0
0.605	99.9
0.720	100.0

G**2+10GSIN(16T)-50COS(32T)
MEAN REM'D OLAP HANNED 19PCS

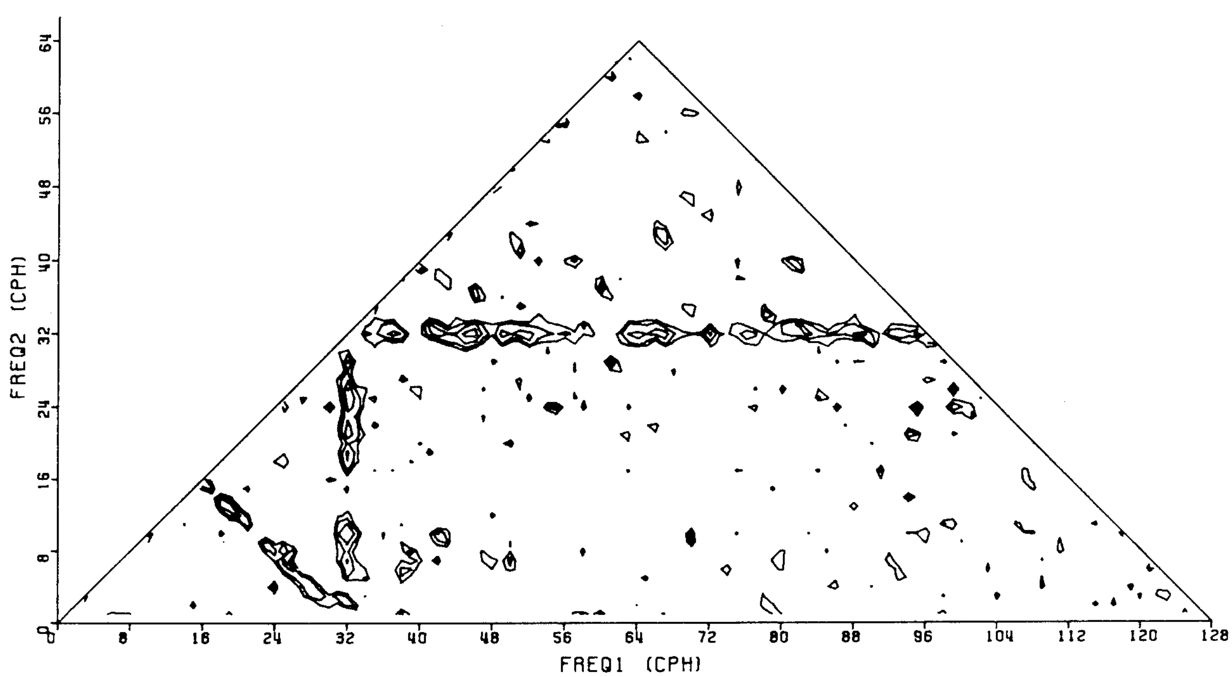


Figure 6

CROSS BICOHERENCE

ORIGINAL FILE NAMES:		VARIABLE NUMBERS:
FREQUENCY 1:	GENRANOUTPUT	0
FREQUENCY 2:	GENRANOUTPUT	0
FREQUENCY 3:	GENRANOUTPUT	0

PROCESSING HISTORY

NO LF'S 4=NO NOS DATA 352=PREW 0=SUBSAMP 0=NSUBSAMP 0=PC SIZE 64
=NO PCS 10=OLAP 1=ISUBT 0=HANN 1 =CREATED 17:18 DEC 15,'77

TIME OF PLOT:
09:38 DEC 22,'77

CONTOUR LEVELS AND PERCENT CONFIDENCE

0.400	80.0
0.431	85.0
0.468	90.0
0.551	95.0
0.675	99.0

X=C8+C16+C24;Y=-S8-S16-S24;FORM XXY
NYQ=32 HANNED S/N=10 FILE BVPL3

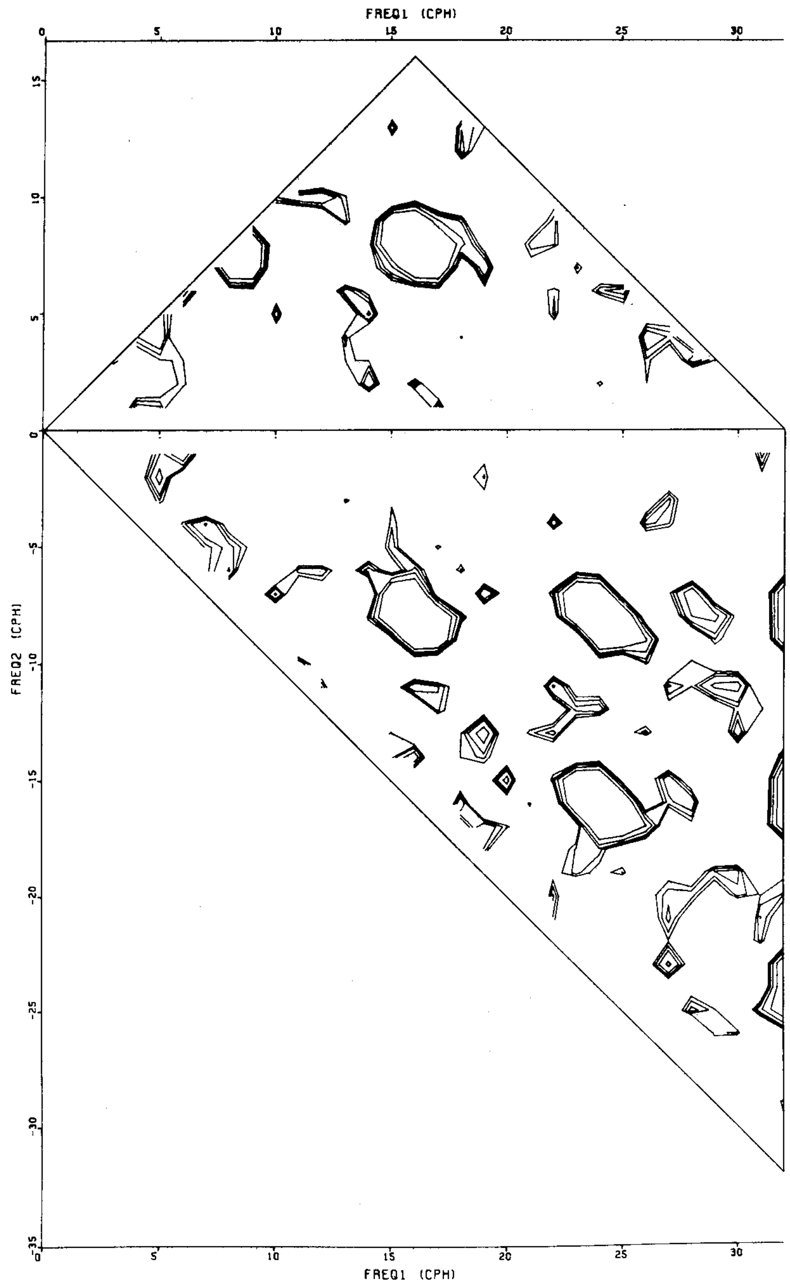


Figure 7

CROSS BICOHERENCE

ORIGINAL FILE NAMES:		VARIABLE NUMBERS:
FREQUENCY 1:	GENRANOUTPUT	0
FREQUENCY 2:	GENRANOUTPUT	0
FREQUENCY 3:	GENRANOUTPUT	0

PROCESSING HISTORY

NO LF'S 8=NO WDS DATA 352=PREW 0=SUBSAMP 0=NSUBSAMP 0=PC SIZE 64
=NO PCS 10=OLAP 1=ISUBT 0=HANN 1 =CREATED 19:20 DEC 21,'77

CONTOUR LEVELS AND PERCENT CONFIDENCE

TIME OF PLOT:
09:28 DEC 22,'77

0.400	80.0
0.431	85.0
0.468	90.0
0.551	95.0
0.675	99.0

X=C8+C16+C24;Y=-S8-S16-S24;FORM XXY
NYQ=32 HANNED UNITIZED S/N=10

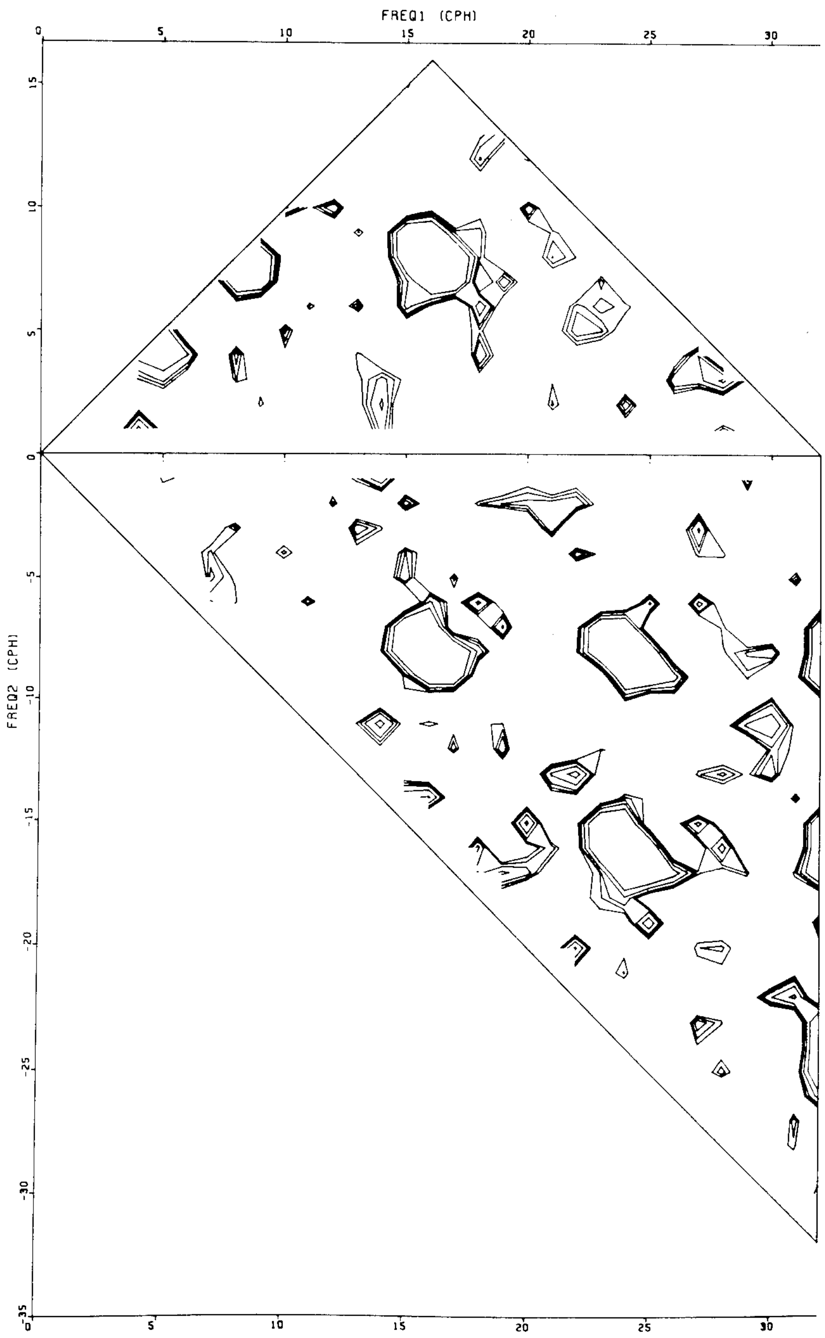


Figure 7a

CROSS BICOHERENCE

ORIGINAL FILE NAMES:		VARIABLE NUMBERS:
FREQUENCY 1:	GENRANOUTPUT	0
FREQUENCY 2:	GENRANOUTPUT	0
FREQUENCY 3:	GENRANOUTPUT	0

PROCESSING HISTORY

NO LF'S 4=NO WDS DATA 352=PREW 0=SUBSAMP 0=NSUBSAMP 0=PC SIZE 64
=NO PCS 10=OLAP 1=ISUBT 0=HANN 1 =CREATED 17:18 DEC 15,'77

TIME OF PLOT:
11:42 DEC 30,'77

CONTOUR LEVELS AND PERCENT CONFIDENCE

0.400	80.0
0.431	85.0
0.468	90.0
0.551	95.0
0.675	99.0

GAUSS NOISE HANNED MEAN=0 STD=1
FORM XXY FILE BVPL1

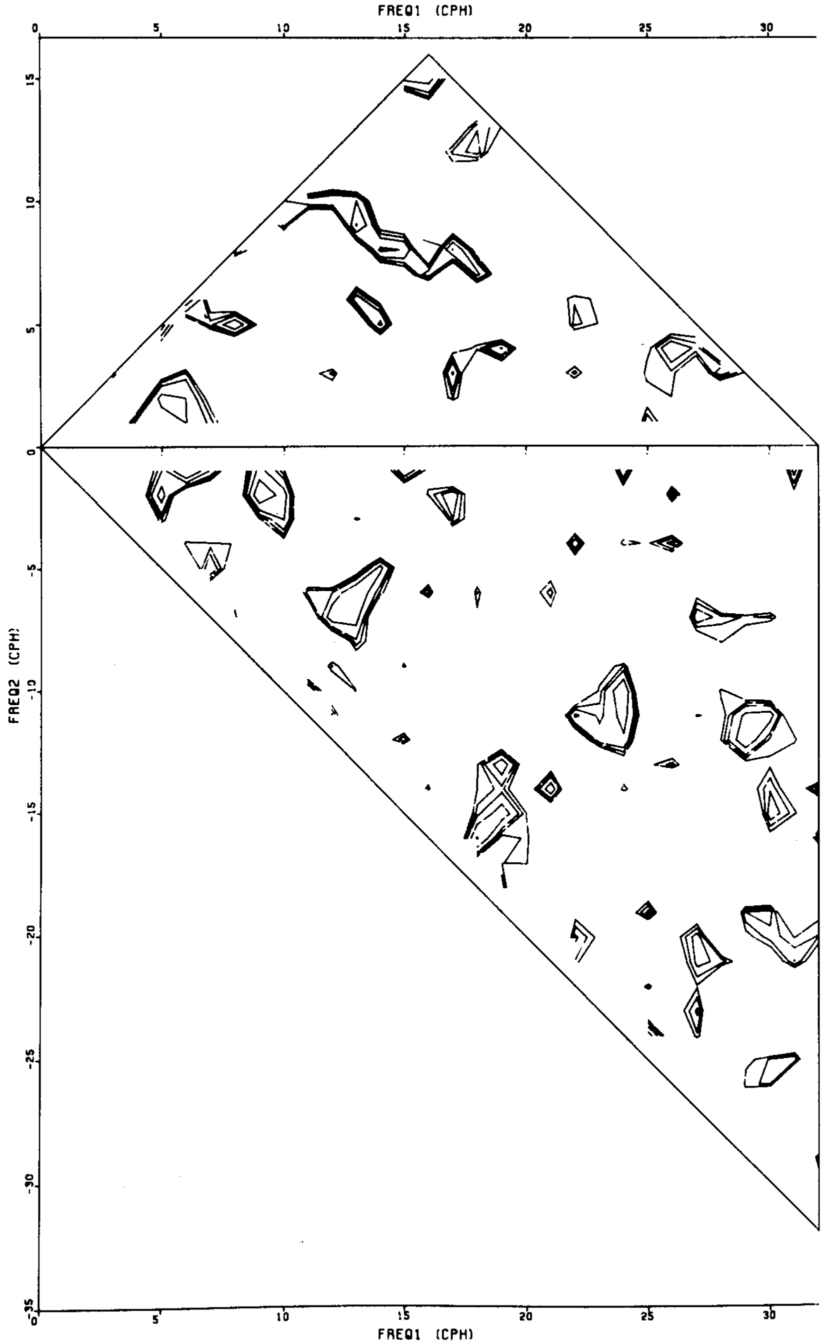


Figure 7b

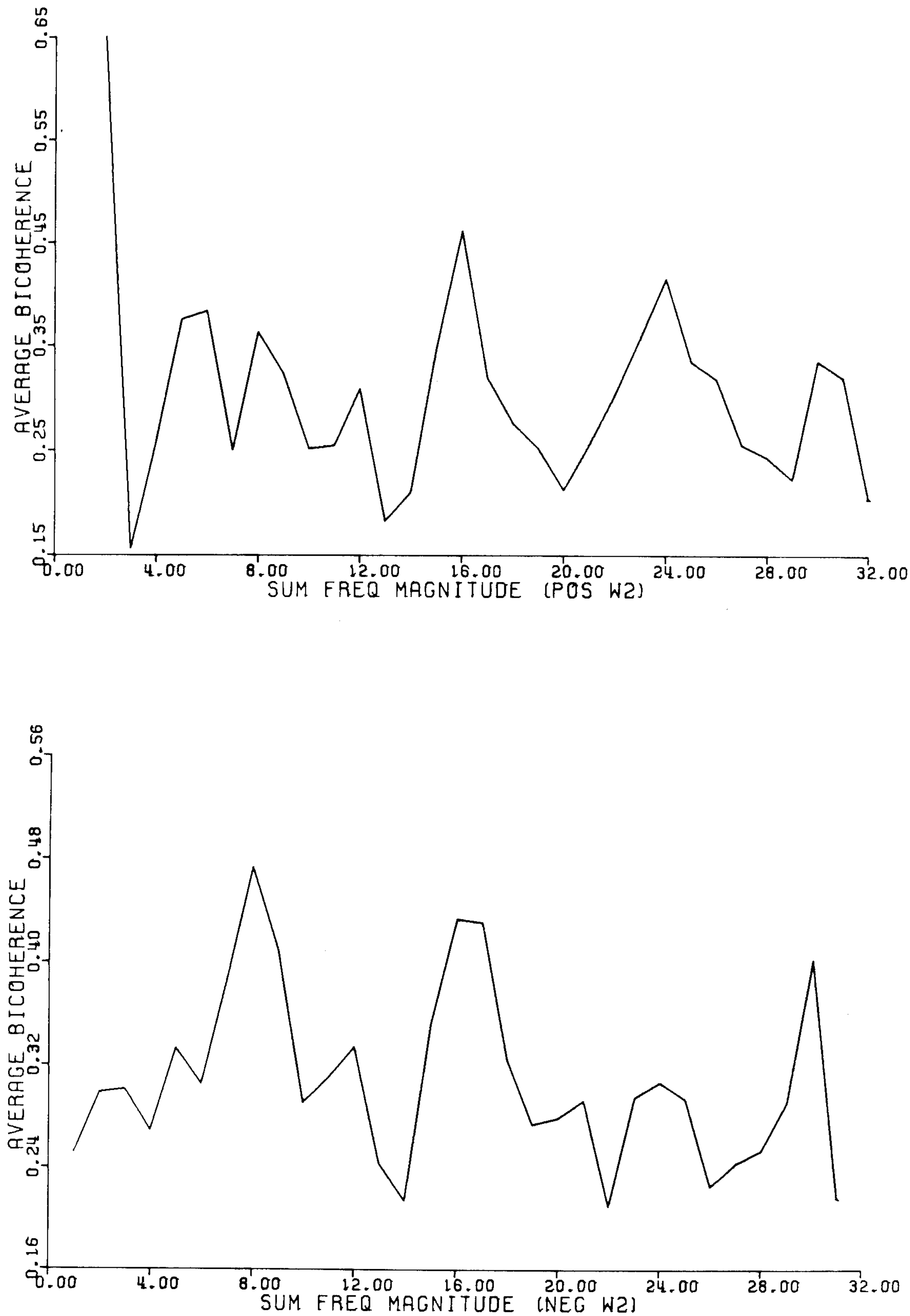
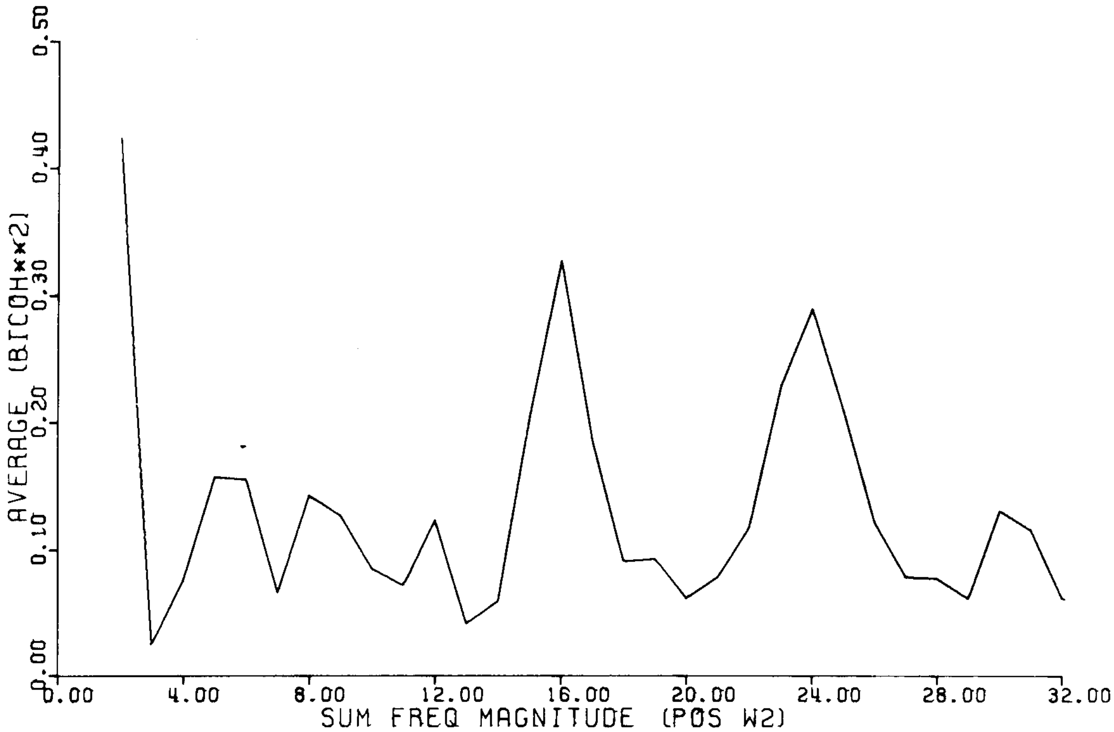

0.65
0.55
0.45
0.35
0.25
0.15
AVERAGE BICOHERENCE
0.00
4.00
8.00
12.00
16.00
20.00
24.00
28.00
32.00
SUM FREQ MAGNITUDE (POS W2)
0.56
0.48
0.40
0.32
0.24
0.16
AVERAGE BICOHERENCE
0.00
4.00
8.00
12.00
16.00
20.00
24.00
28.00
32.00
SUM FREQ MAGNITUDE (NEG W2)

Figure 7c


0.50
0.40
0.30
0.20
0.10
0.00
AVERAGE (BICOH**2)
0.00
4.00
8.00
12.00
16.00
20.00
24.00
28.00
32.00
SUM FREQ MAGNITUDE (POS W2)

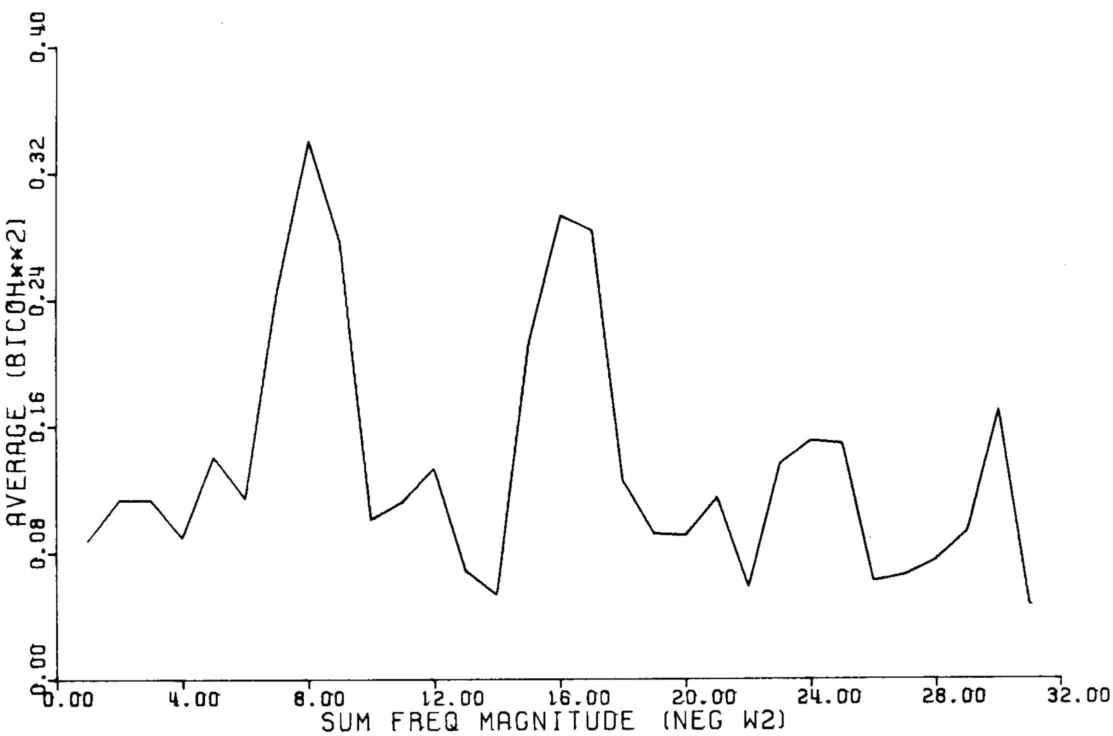

0.40
0.32
0.24
0.16
0.08
0.00
AVERAGE (BICOH**2)
0.00
4.00
8.00
12.00
16.00
20.00
24.00
28.00
32.00
SUM FREQ MAGNITUDE (NEG W2)

Figure 7d

AUTO ROTARY BICOHERENCE

CONTOUR LEVELS AND PERCENT CONFIDENCE

0.406	80.000
0.436	85.000
0.481	90.000
0.565	95.000
0.715	99.000

GAUSSIAN NOISE MEAN=0. S.D.=1
YVEC,YV1,YV2,YVPL1

PROCESSING HISTORY

NO LF'S 11*NO WDS DATA 352*PREW 0*
SUBSAMP 0*NSUBSAMP 0*PC SIZE 64*NO PCS 10*
OLAP 1*ISUBT 0*HANN 1 *CREATED 14:28 DEC 07,'77

TIME OF PLOT:
14:52 DEC 07,'77

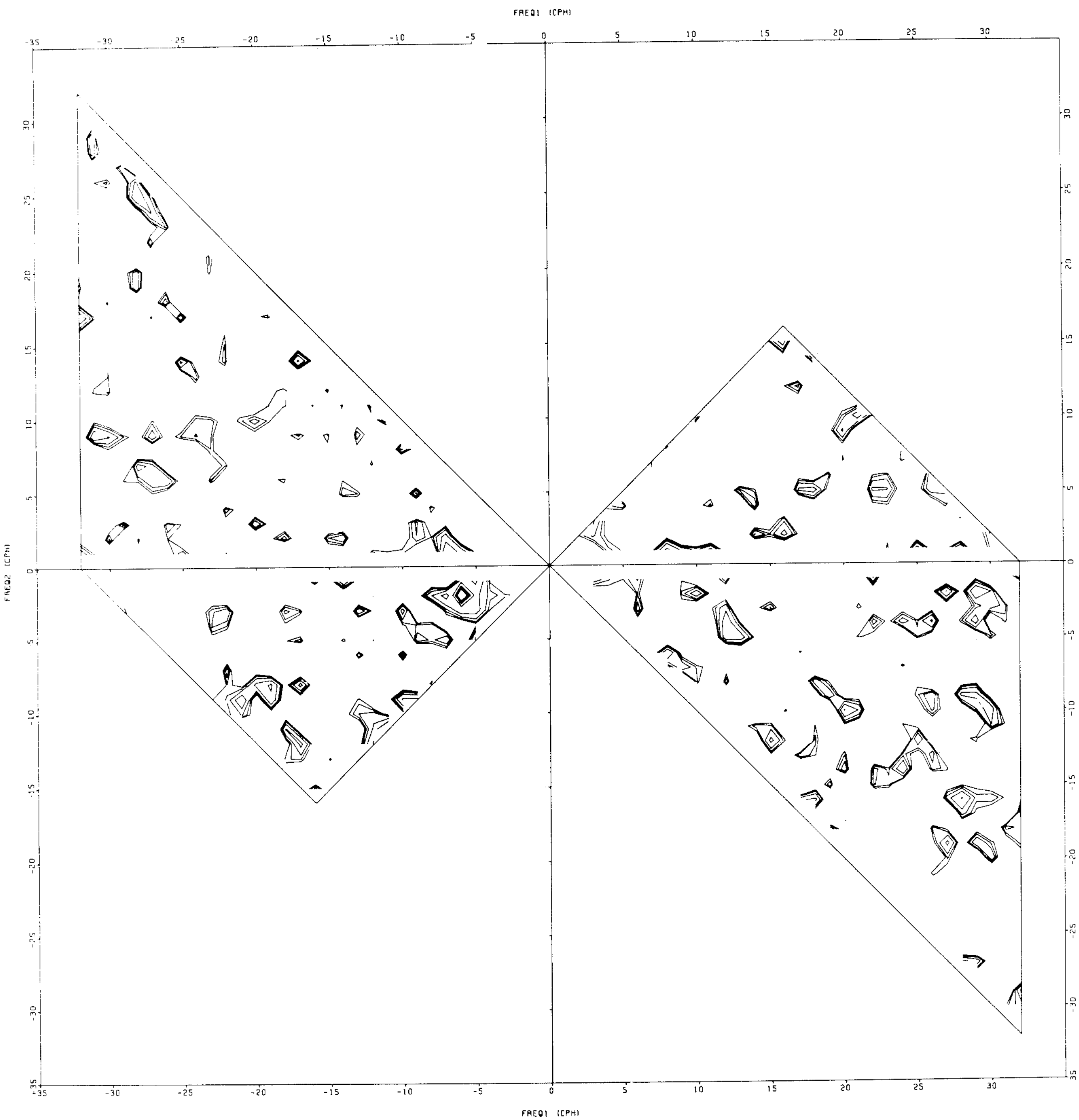


Figure 8

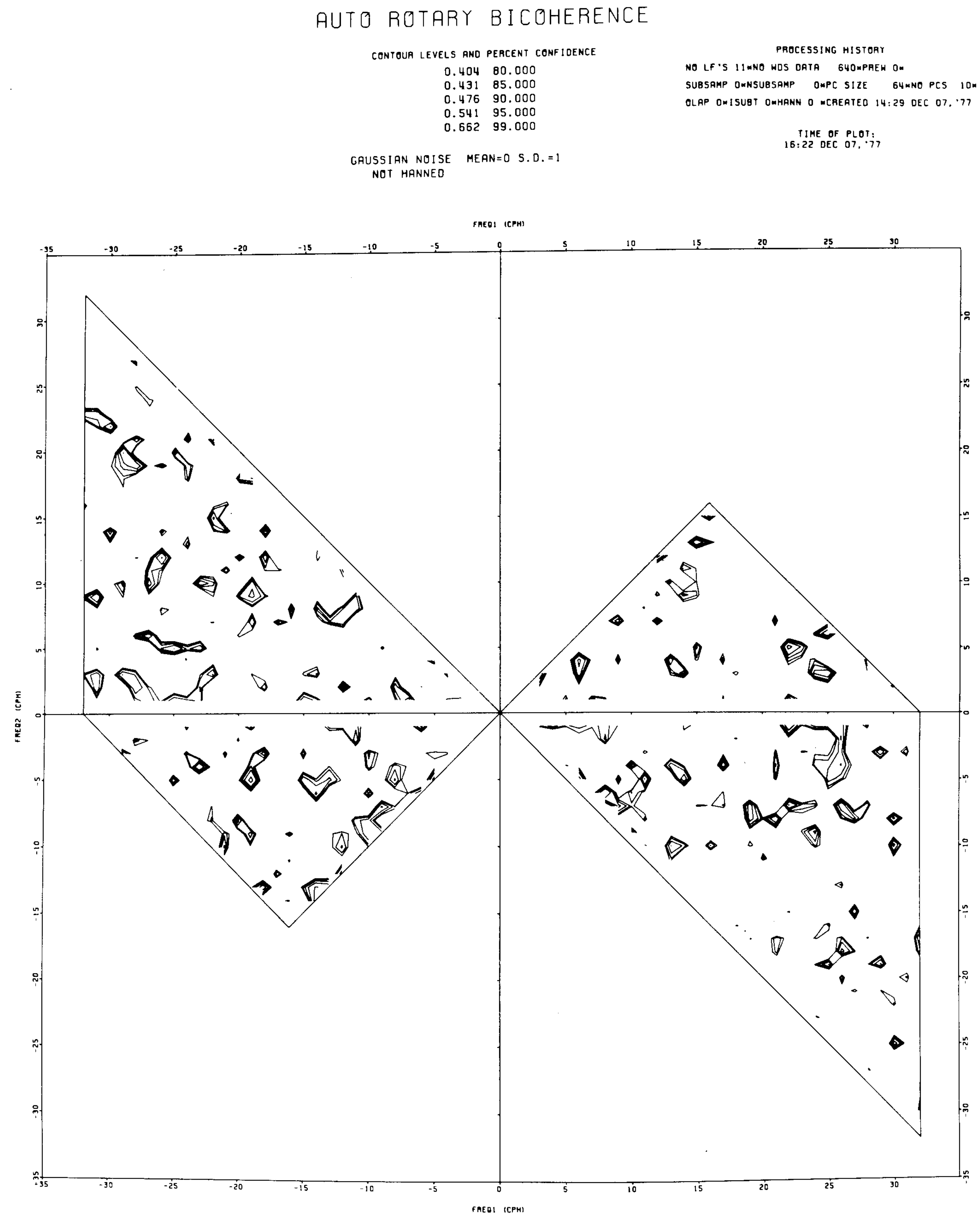

AUTO ROTARY BICOHERENCE
CONTOUR LEVELS AND PERCENT CONFIDENCE
0.404 80.000
0.431 85.000
0.476 90.000
0.541 95.000
0.662 99.000
GAUSSIAN NOISE MEAN=0 S.D.=1
NOT HANNED
PROCESSING HISTORY
NO LF'S 11*NO WDS DATA 640*PREW 0*
SUBSAMP 0*NSUBSAMP 0*PC SIZE 64*NO PCS 10*
OLAP 0*ISUBT 0*HANN 0 *CREATED 14:29 DEC 07,'77
TIME OF PLOT:
16:22 DEC 07,'77
FREQ1 (CPH)
FREQ2 (CPH)
-35 -30 -25 -20 -15 -10 -5 0 5 10 15 20 25 30
30 25 20 15 10 5 0 -5 -10 -15 -20 -25 -30 -35

Figure 9

AUTO ROTARY BICOHERENCE

CONTOUR LEVELS AND PERCENT CONFIDENCE

0.406	80.000
0.436	85.000
0.481	90.000
0.565	95.000
0.715	99.000

3 COUNTER RCT VEC'S S/N 10 HANNED
FREQ 8,16,24 NYQUIST=32

PROCESSING HISTORY
NO LF'S 11*NO WCS DATA 352*PREW 0*
SUBSAMP 0*NSUBSAMP 0*PC SIZE 64*NO PCS 10*
OLAP 1*JSUBT 0*HANN 1 *CREATED 14:28 DEC 07,'77

TIME OF PLOT:
15:43 DEC 07,'77

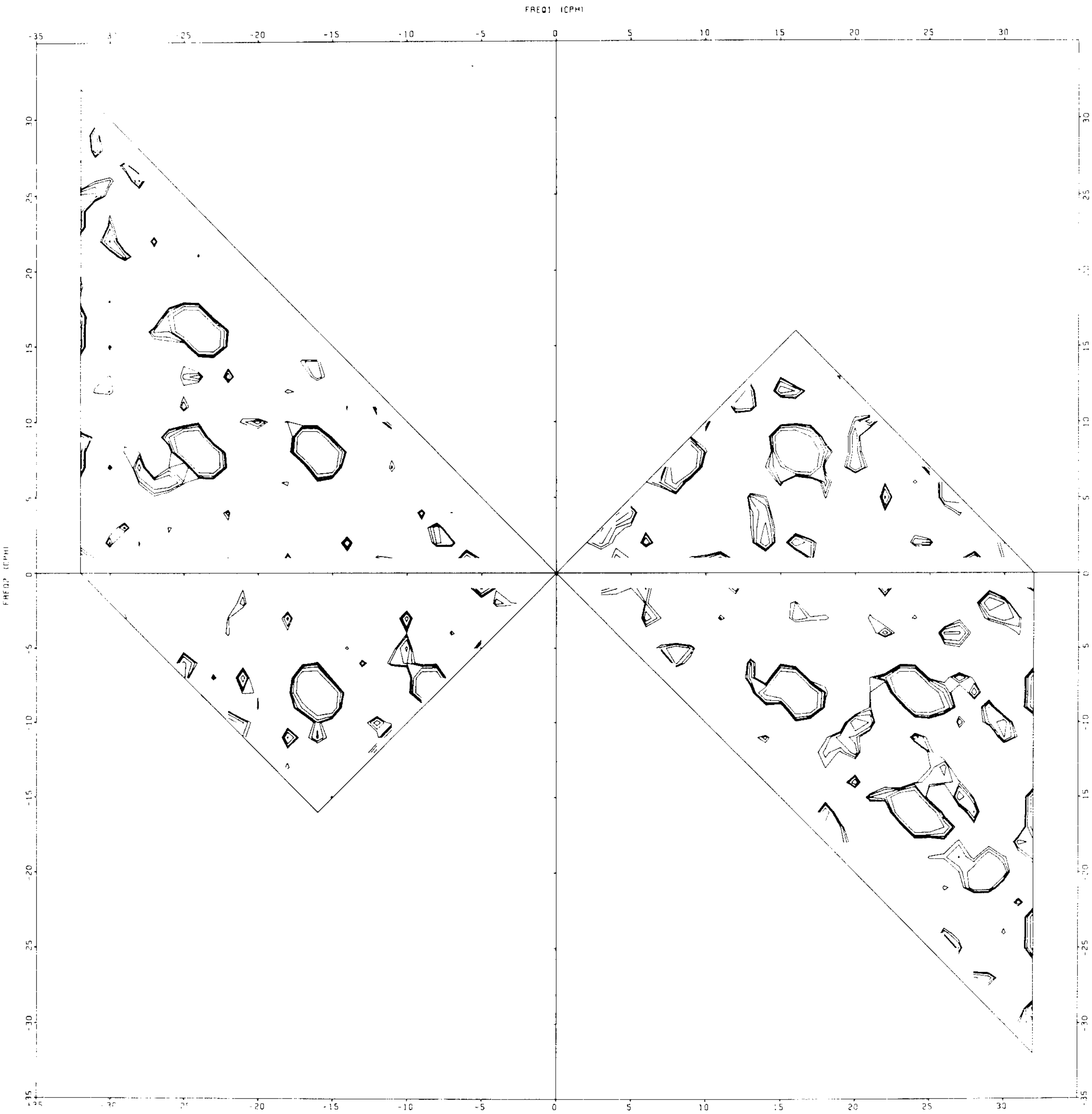


Figure 10

AUTO ROTARY BICOHERENCE

CONTOUR LEVELS AND PERCENT CONFIDENCE

0.406	80.000
0.436	85.000
0.481	90.000
0.565	95.000
0.715	99.000

3 COUNTER ROT VEC'S S/N=1 HANNED
FREQ 8,16,24 NYQUIST=32

PROCESSING HISTORY
NO LF'S 11*NO WDS DATA 352*PREW 0*
SUBSAMP 0*NSUBSAMP 0*PC SIZE 64*NO PCS 10*
OLAP 1*ISUBT 0*HANN 1 *CREATED 14:28 DEC 07,'77

TIME OF PLOT:
15:48 DEC 07,'77

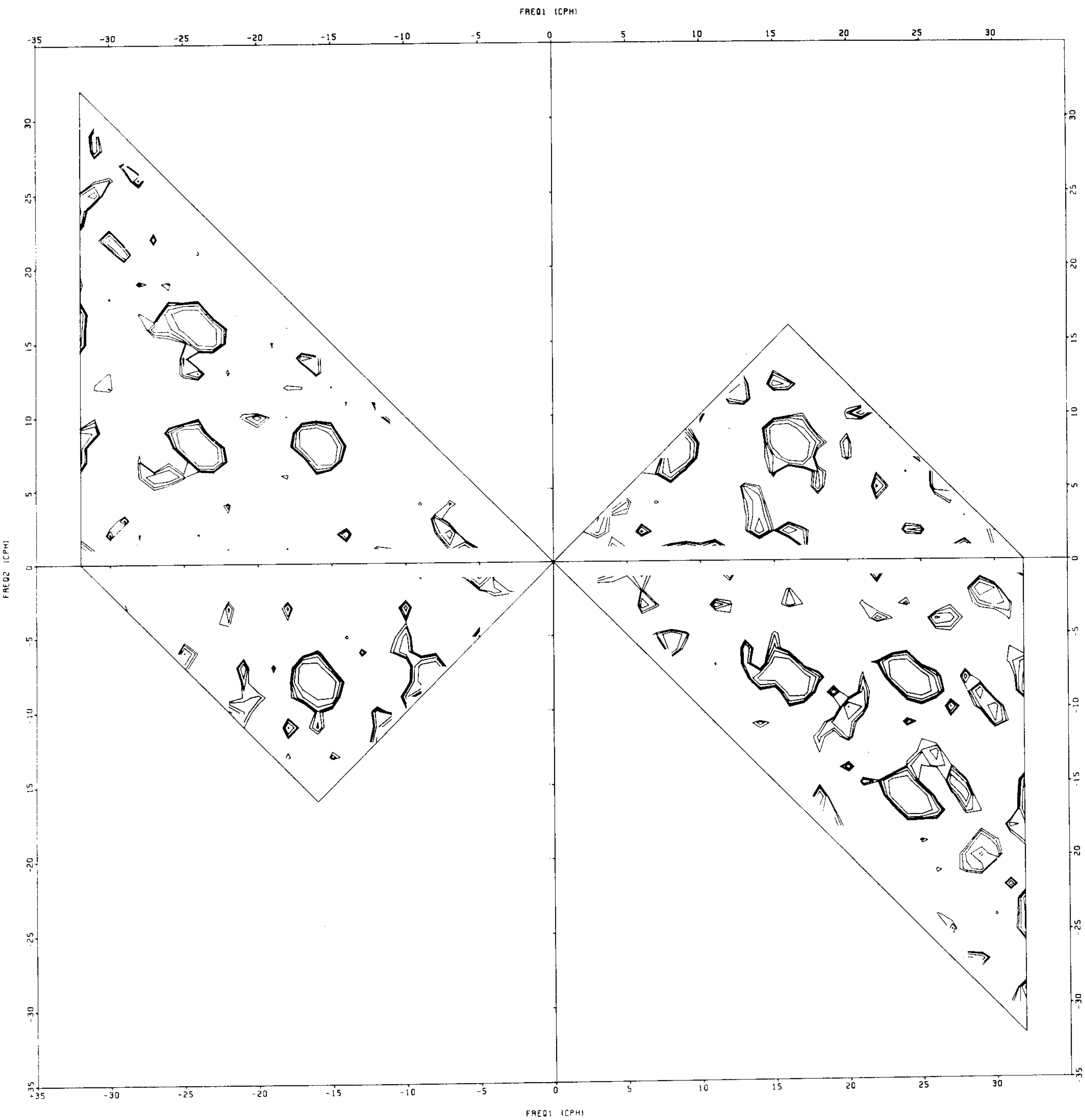


Figure 11

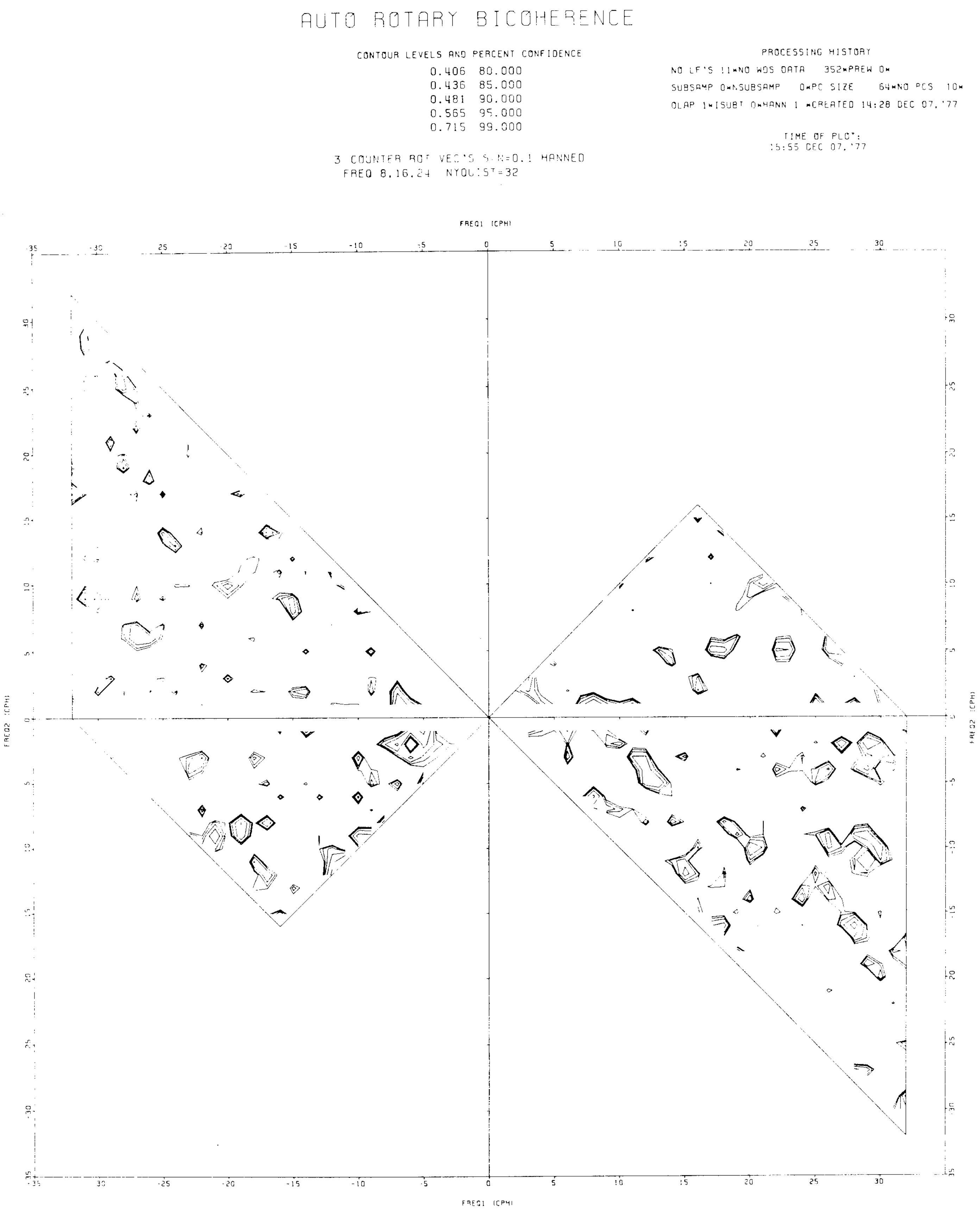
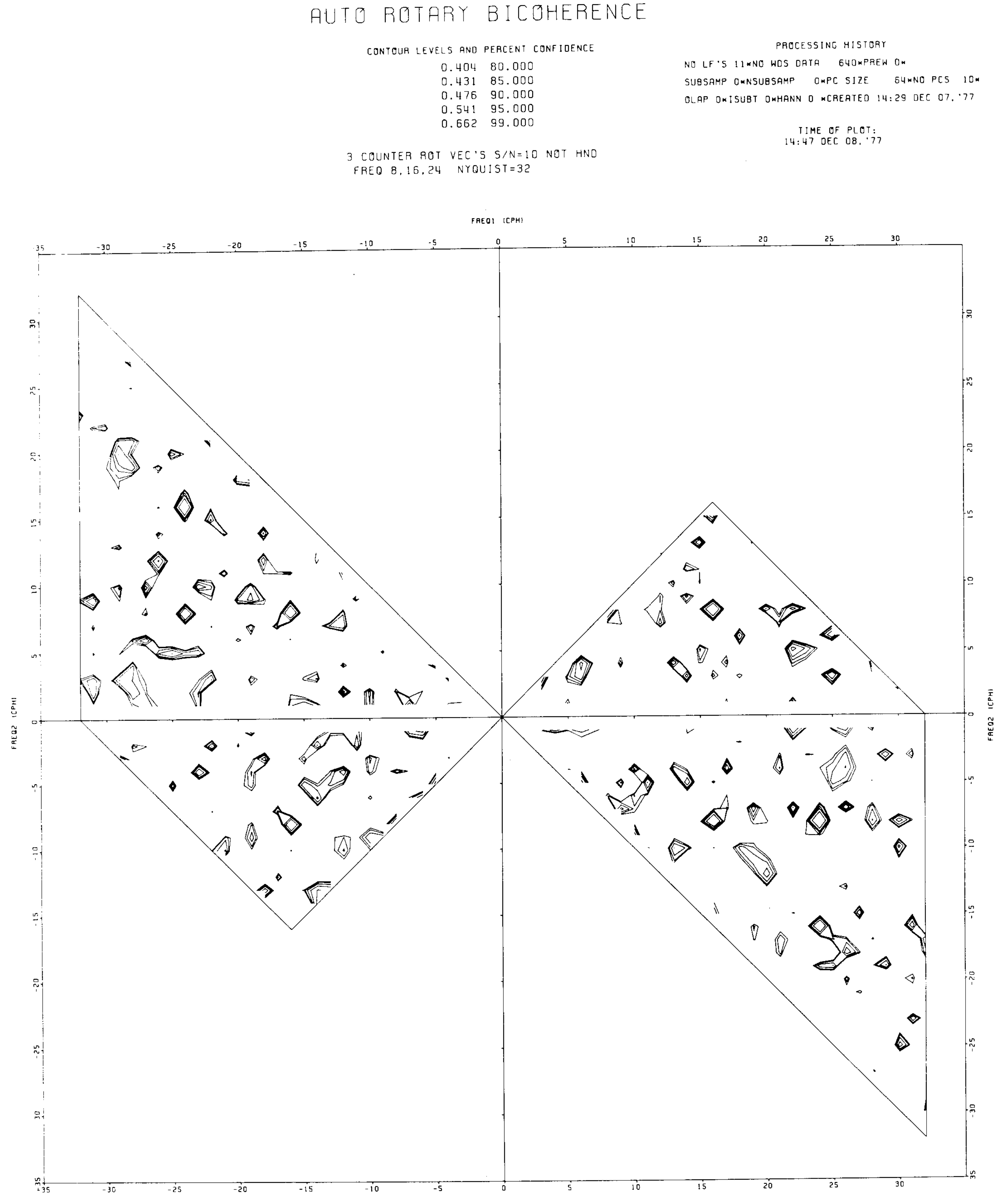

AUTO ROTARY BICOHERENCE
CONTOUR LEVELS AND PERCENT CONFIDENCE
0.406 80.000
0.436 85.000
0.481 90.000
0.565 95.000
0.715 99.000
3 COUNTER ROT VEC'S S:N=0.1 HANNED
FREQ 8,16,24 NYQUIST=32
PROCESSING HISTORY
NO LF'S 11*NO WDS DATA 352*PREW 0*
SUBSAMP 0*NSUBSAMP 0*PC SIZE 64*NO PCS 10*
OLAP 1*ISUBT 0*HANN 1 *CREATED 14:28 DEC 07,'77
TIME OF PLOT:
15:55 DEC 07,'77
FREQ1 (CPH)
FREQ2 (CPH)

Figure 12



AUTO ROTARY BICOHERENCE
CONTOUR LEVELS AND PERCENT CONFIDENCE
0.404 80.000
0.431 85.000
0.476 90.000
0.541 95.000
0.662 99.000
3 COUNTER ROT VEC'S S/N=10 NOT HND
FREQ 8,16,24 NYQUIST=32
PROCESSING HISTORY
NO LF'S 11*NO WDS DATA 640*PREW 0*
SUBSAMP 0*NSUBSAMP 0*PC SIZE 64*NO PCS 10*
OLAP 0*ISUBT 0*HANN 0 *CREATED 14:29 DEC 07,'77
TIME OF PLOT:
14:47 DEC 08,'77
FREQ1 (CPH)
FREQ2 (CPH)

Figure 13

AUTO ROTARY BICOHERENCE

CONTOUR LEVELS AND PERCENT CONFIDENCE

0.406	80.000
0.436	85.000
0.481	90.000
0.565	95.000
0.715	99.000

3 CW VEC'S S/N=10 HANNED
FREQ 8,16,24 NYQUIST=32

PROCESSING HISTORY
NO LF'S 4*NO WDS DATA 352*PREW 0*
SUBSAMP 0*NSUBSAMP 0*PC SIZE 64*NO PCS 10*
OLAP 1*ISUBT 0*HANN 1 *CREATED 17:18 DEC 15,'77

TIME OF PLOT:
10:57 DEC 16,'77

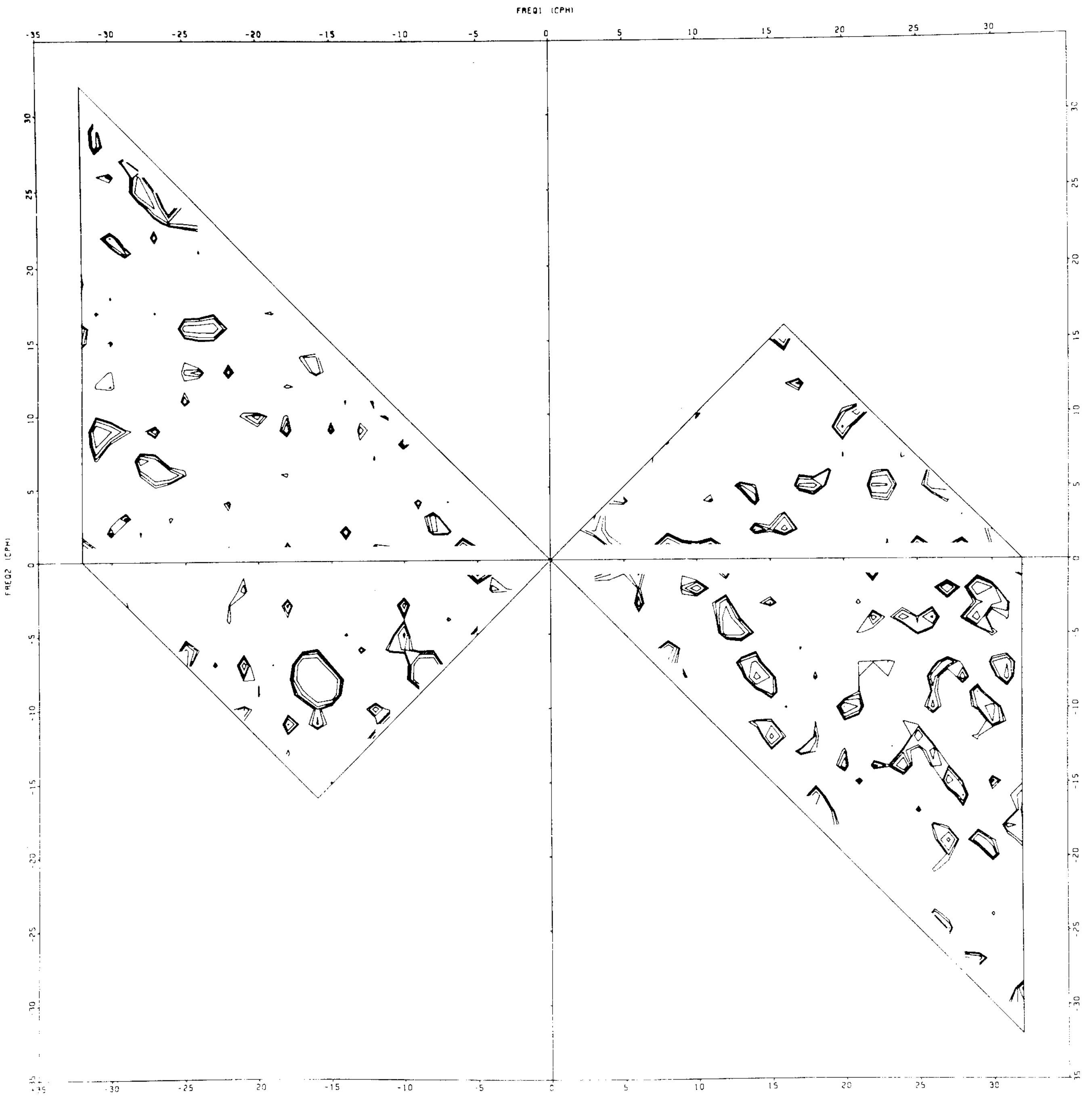


Figure 14

AUTO ROTARY BICOHERENCE

CONTOUR LEVELS AND PERCENT CONFIDENCE

0.406	80.000
0.436	85.000
0.481	90.000
0.565	95.000
0.715	99.000

3 CCW VEC'S S/N=10 HANNED
FREQ 8,16,24 NYQUIST=32

PROCESSING HISTORY
NO LF'S 11*NO WDS DATA 352*PREW 0*
SUBSAMP 0*NSUBSAMP 0*PC SIZE 64*NO PCS 10*
OLAP 1*ISUBT 0*HANN 1 *CREATED 14:28 DEC 07,'77

TIME OF PLOT:
15:00 DEC 08,'77

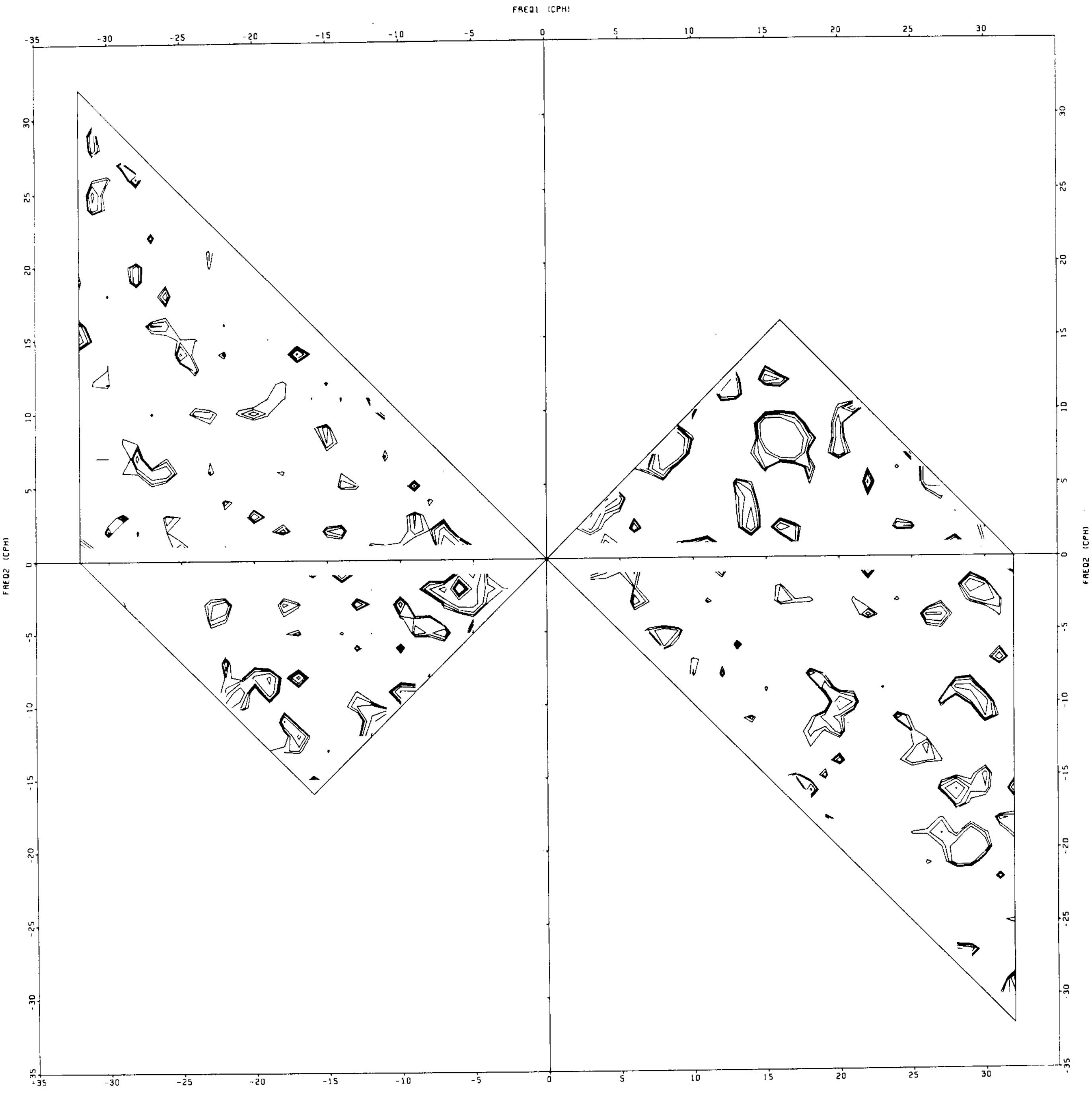


Figure 15

AUTO ROTARY BICOHERENCE

CONTOUR LEVELS AND PERCENT CONFIDENCE

0.541	95.000
0.556	96.000
0.579	97.000
0.616	98.000
0.662	99.000

GAUSSIAN NOISE MEAN=0, S.D.=1
NOT HANNED

PROCESSING HISTORY
NO LF'S 1*NO WDS DATA 640*PREW 0*
SUBSAMP 0*NSUBSAMP 0*PC SIZE 64*NO PCS 10*
OLAP 0*ISUBT 0*HANN 0 *CREATED 14:29 DEC 07,'77

TIME OF PLOT:
13:55 DEC 08,'77

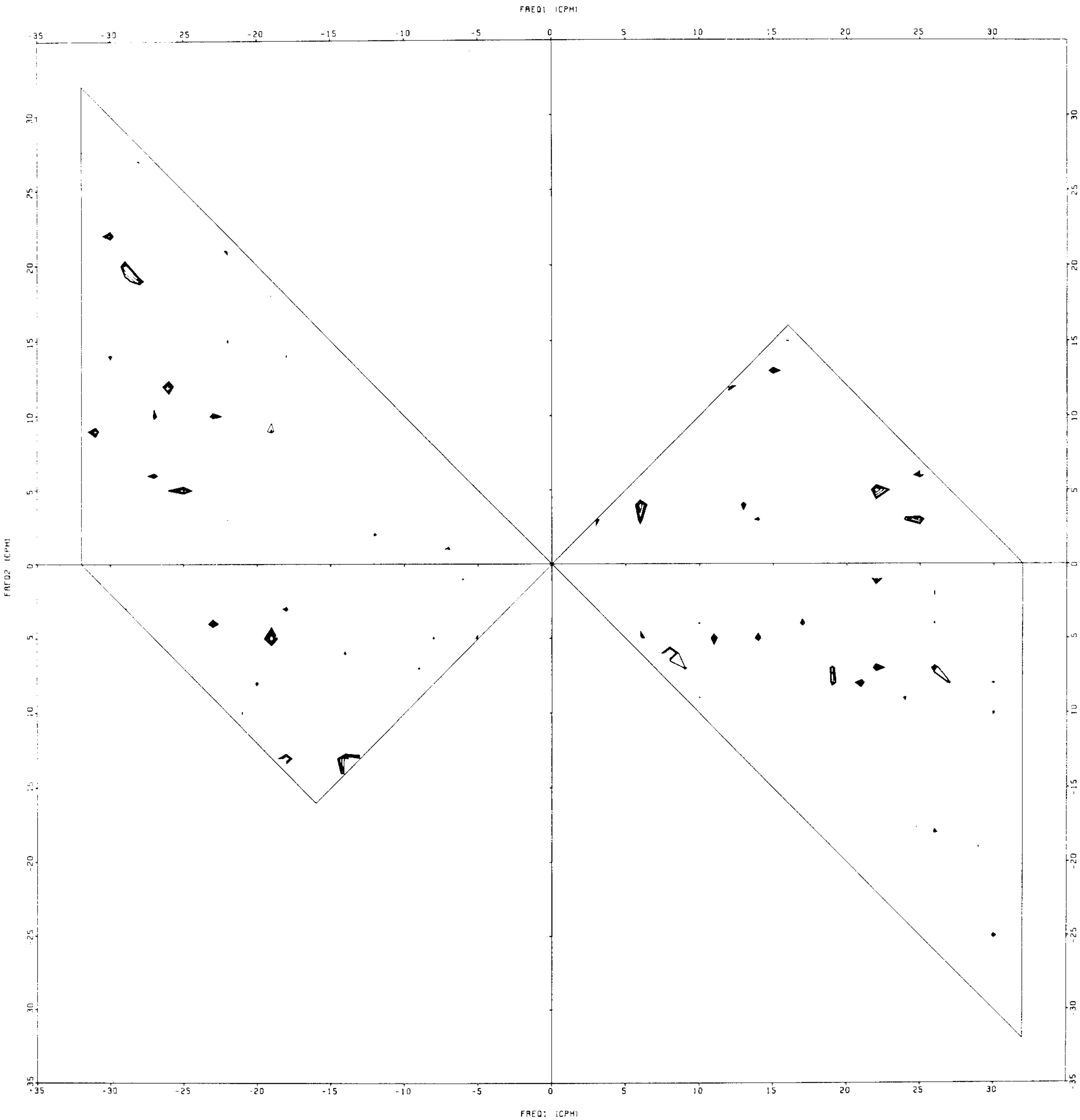


Figure 16

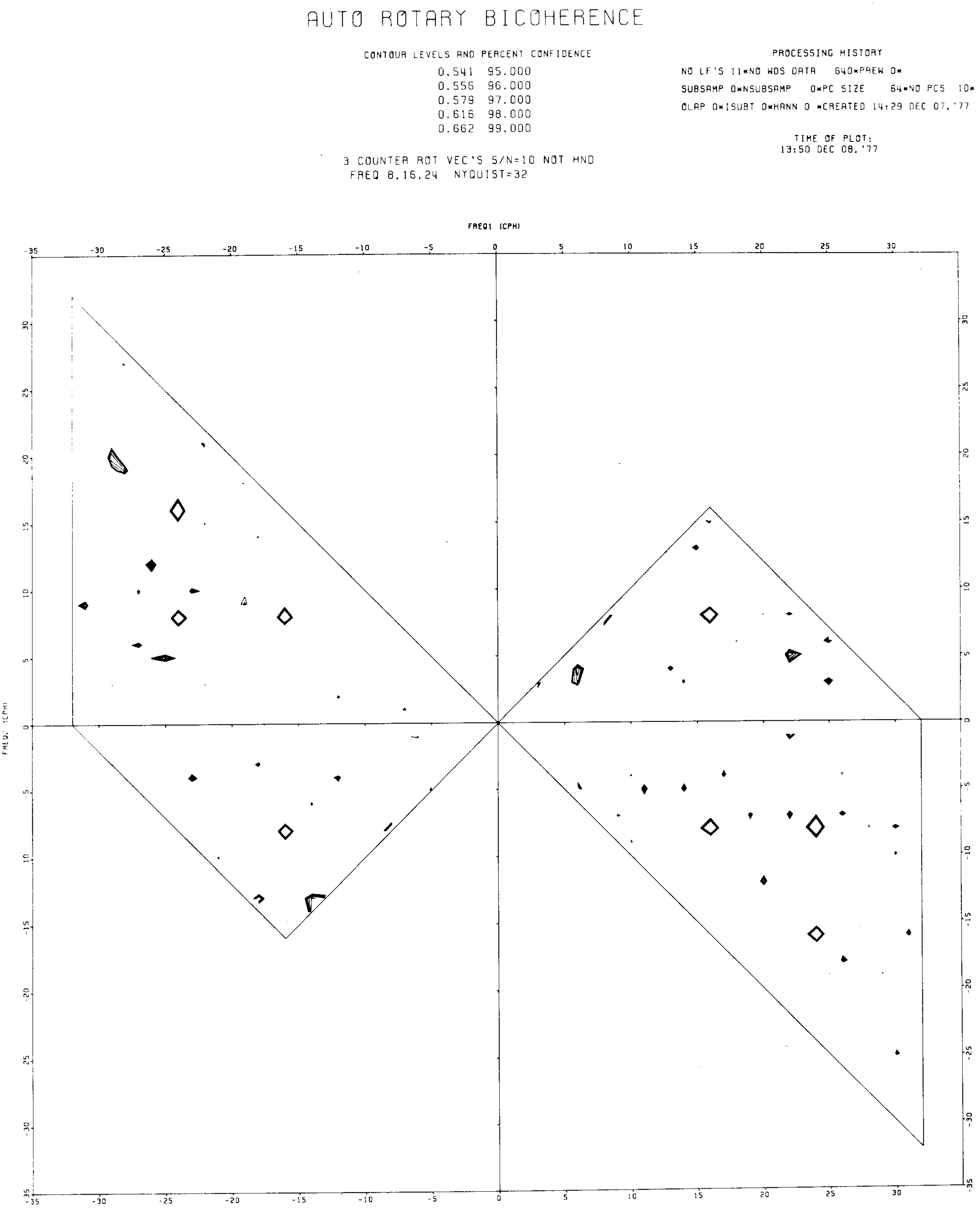

AUTO ROTARY BICOHERENCE
CONTOUR LEVELS AND PERCENT CONFIDENCE
0.541 95.000
0.556 96.000
0.579 97.000
0.616 98.000
0.662 99.000
3 COUNTER ROT VEC'S S/N=10 NOT HND
FREQ 8,16,24 NYQUIST=32
PROCESSING HISTORY
NO LF'S 11*NO HDS DATA 640*PREW 0*
SUBSAMP 0*NSUBSAMP 0*PC SIZE 64*NO PCS 10*
OLAP 0*ISUBT 0*HANN 0 *CREATED 14:29 DEC 07,'77
TIME OF PLOT:
13:50 DEC 08,'77
FREQ1 (CPH)
FREQ2 (CPH)

Figure 17

CROSS ROTARY BICOHERENCE

CONTOUR LEVELS AND PERCENT CONFIDENCE

0.406	80.000
0.436	85.000
0.481	90.000
0.565	95.000
0.715	99.000

S/N=10 HANNED SER 1: -8,-16,16
SER 2: 8,16,-24 NYQUIST=32

PROCESSING HISTORY

NO LF'S 11*NO WDS DATA 352*PREW 0*
SUBSAMP 0*NSUBSAMP 0*PC SIZE 64*NO PCS 10*
OLAP 1*ISUBT 0*HANN 1 *CREATED 14:28 DEC 07,'77

TIME OF PLOT:
16:14 DEC 07,'77

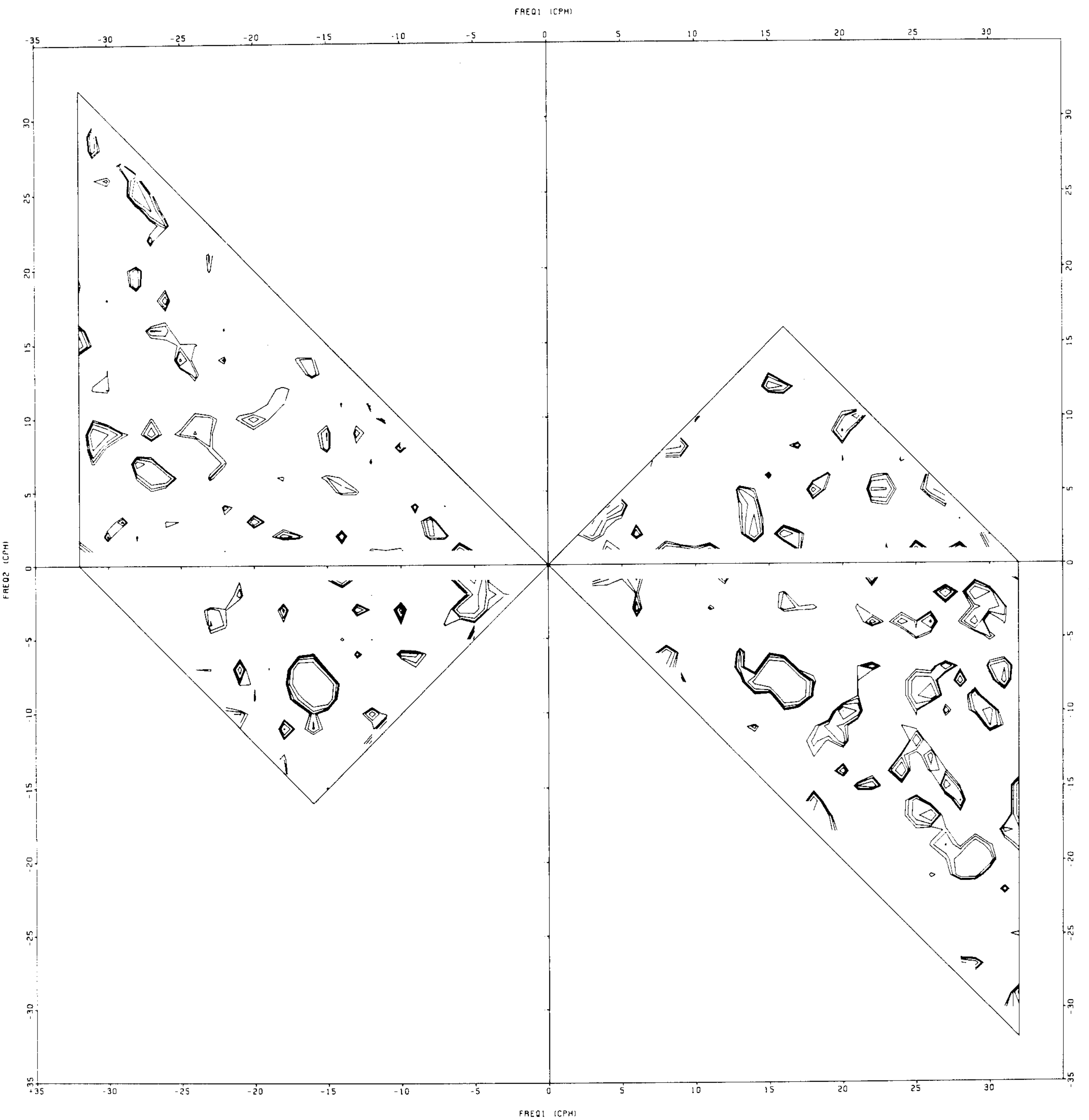


Figure 18

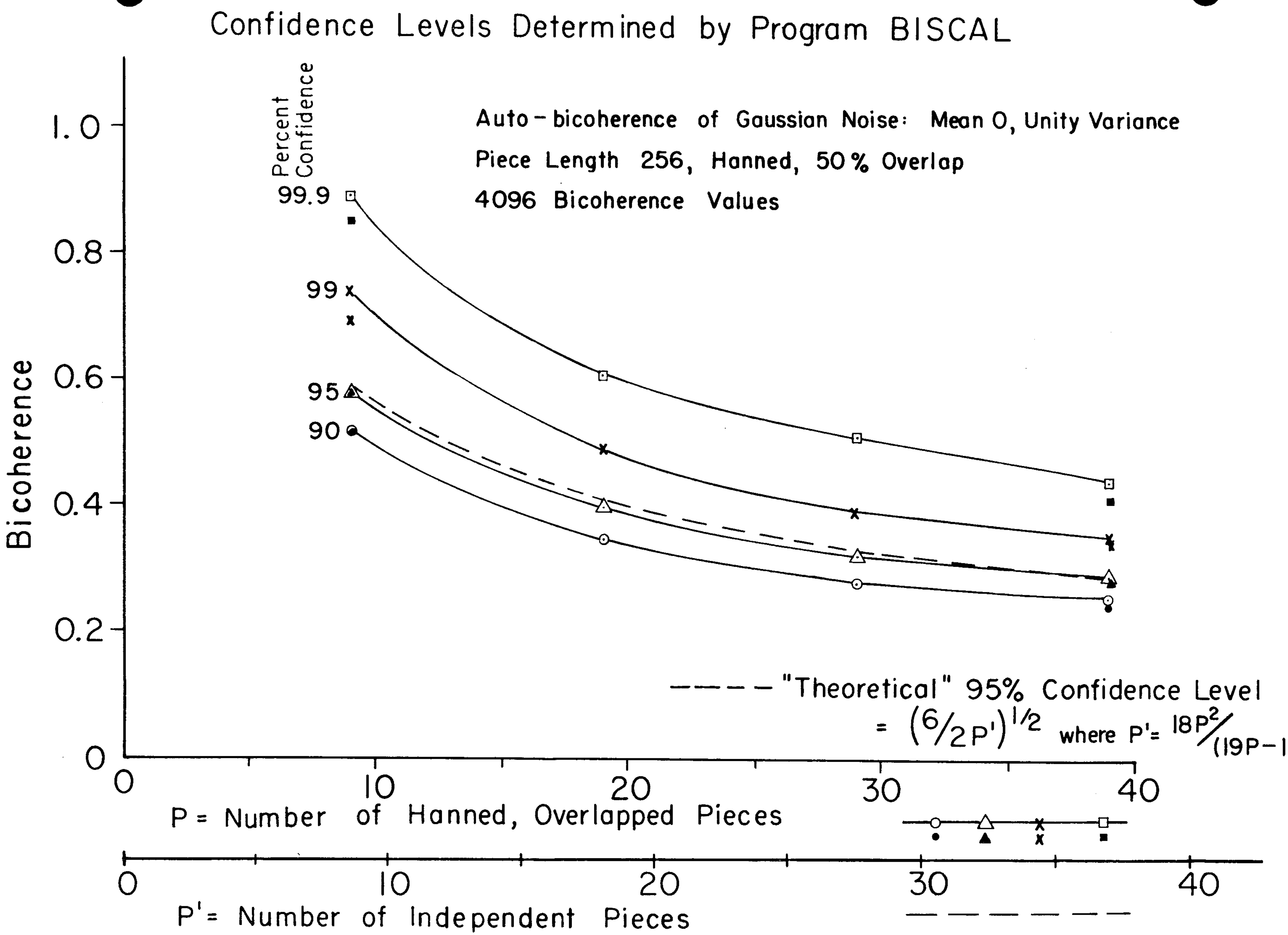

Confidence Levels Determined by Program BISCAL
Auto-bicoherence of Gaussian Noise: Mean 0, Unity Variance
Piece Length 256, Hanned, 50% Overlap
4096 Bicoherence Values
Percent Confidence
99.9
99
95
90
Bicoherence
1.0
0.8
0.6
0.4
0.2
0
"Theoretical" 95% Confidence Level
$= (6/2P')^{1/2}$ where $P' = 18P^2/(19P-1)$
0
10
20
30
40
P = Number of Hanned, Overlapped Pieces
P' = Number of Independent Pieces

Figure 19

II. Bispectral Programs

A. Introduction To the Bispectral Program Reports

1. Summary

Following is a brief summary of the tasks performed by the programs described in Sections B, C, and D below. In reviewing this summary it will be helpful to refer to the flowchart presented immediately following the summary in Section II.A.2.

Building RWDISC Data File Preparatory to Fourier Transform (Section B)

a. FRAGTAP; ORDAT: Two main programs, run consecutively from TAPDIS (RWDISC) file having multiple pseudo-files to a different RWDISC file having 21 logical files, the 21st of which is a facsimile of TAPDIS pseudo-file 1 (labels)

b. GENRAN:* Main program which generates RWDISC file of series containing any combination of:

1) Gaussian noise

2) non-Gaussian noise of form:

$$ax^2 + bx^3$$

where x is Gaussian noise

3) signals of form:

$$\sum_{i=1}^{4} A_1 \sin(f_{i1}t+\phi_{i1})\sin(f_{i2}t+\phi_{i2})$$

4) harmonics of form:

$$\sum_{i=1}^{6} A_i \sin(ift+\phi_i)$$

Modifying RWDISC File from I by Replacing Data Series with Fourier Coefficients in a Special Order (Section C)

a. FOURIER: Main program which does Fourier transform, including options of prewhitening, triangular weighting of consecutive subsamples, splitting into contiguous or overlapping pieces, subtraction of mean or trend, and Hanning.

Bispectral Calculations and Plotting (Section D)

a. BISCAL: Works from specially organized RWDISC file produced by program FOURIER, to calculate auto and cross-bispectra of two real series, to write a disc file of bicoherences for transmittal to plotting program, BPLOT, and to compute confidence levels for the bicoherences.

b. BPLOT: Works from disc file produced by programs BISCAL or GENRAN to plot auto and cross-bicoherences.

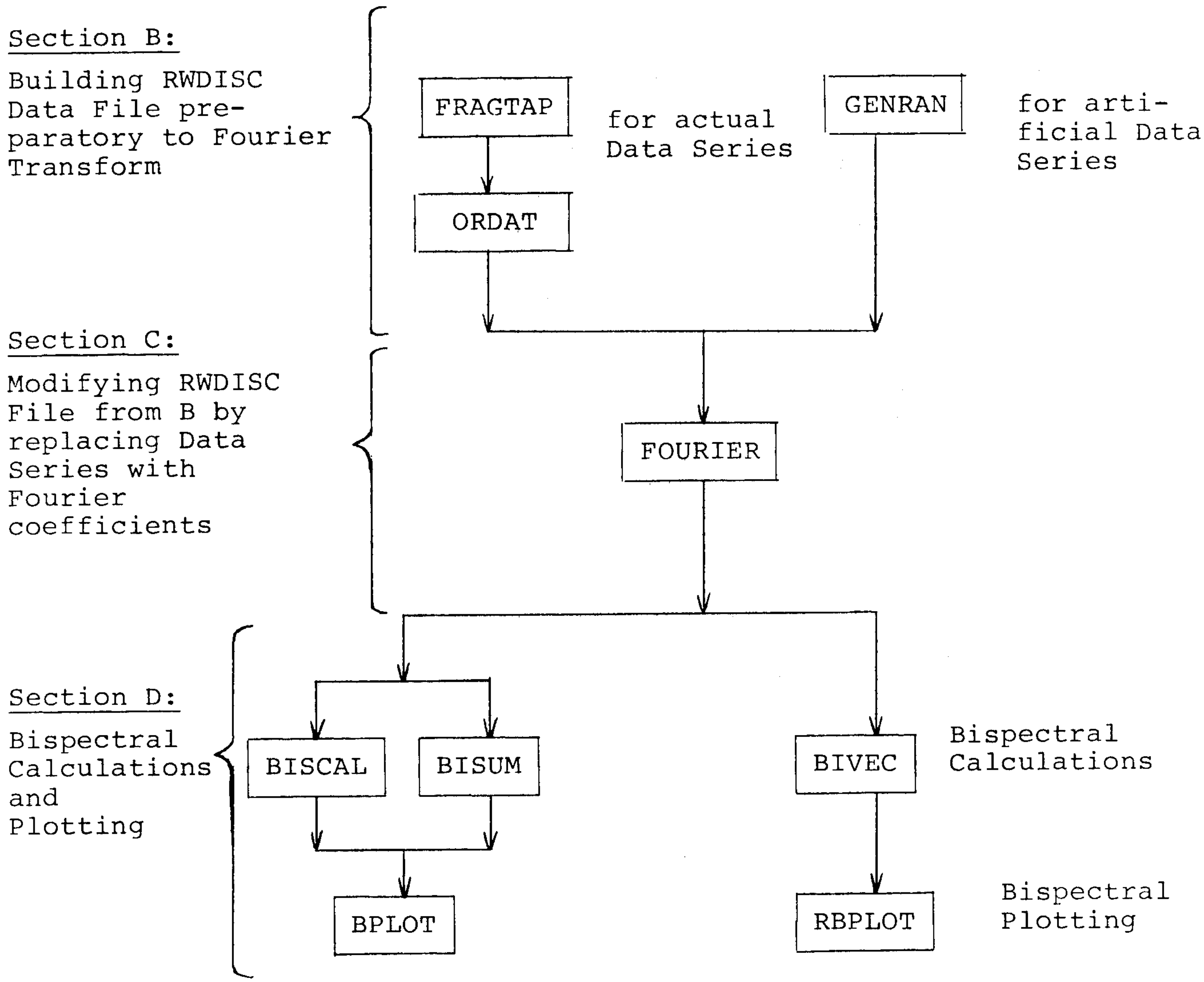
c. BISUM: Works from RWDISC file produced by program FOURIER to integrate certain bispectral quantities along paths of constant sum frequency in the ω_1,ω_2 plane. Produces disc file containing integrals, as well as another disc file for input into plotting program BPLOT.

d. BIVEC: Works from RWDISC FILE produced by program FOURIER, to calculate auto and cross rotary bispectra of two vector series, to write disc files of rotary bicoherences for transmittal to plotting program RBPLOT, and to compute confidence levels for the rotary bicoherences.

e. RBPLOT: Works from disc files produced by programs BIVEC or GENRAN to make contour plots of auto and cross rotary bicoherences.

2. Flowchart for the Bispectral Programs

Sections IIB, C, D, and E contain reports and listings for nine programs connected with bispectral analysis. The following flowchart is presented as an aid to understanding the functions of the various programs, and should be consulted in reviewing the reports on them:


Section B:
Building RWDISC Data File preparatory to Fourier Transform
FRAGTAP
for actual Data Series
GENRAN
for artificial Data Series
ORDAT
Section C:
Modifying RWDISC File from B by replacing Data Series with Fourier coefficients
FOURIER
Section D:
Bispectral Calculations and Plotting
BISCAL
BISUM
BIVEC
Bispectral Calculations
BPLOT
RBPLOT
Bispectral Plotting

3. Compatibility with Other Systems

The programs described in the reports and listings in Section II are all system-dependent, the system being the Xerox CP-V system on the XEROX SIGMA-7 computer at Woods Hole Oceanographic Institution (WHOI). This is particularly true of those programs which retrieve and organize the data: FRAGTAP, ORDAT, and FOURIER. If one were to attempt to adapt any of the programs described below to another system, the most tedious but straightforward task would be to convert the Xerox Extended Fortran-IV statements used by these programs to new Fortran-IV statements wherever necessary. A helpful discussion of differences between Xerox Extended Fortran-IV and ANS Standard Fortran is given in the reference manual for the former.[1] One would next have to turn attention to the subroutines used by these programs, and which reside in the system library at WHOI. These subroutines are not generally listed in the reports. It would be necessary to either rewrite the subroutines to be compatible with the new system, or replace them by equivalent subroutines. The following points would be helpful:

(a) Mention is often made of "RWDISC" data files throughout these reports. RWDISC is a machine-language program written at WHOI to provide users with a convenient method of using the disc as a true random-access file(s). It uses FORTRAN-callable entry points for reading, writing, and initializing, and it sets up a <u>conceptually</u> simple disc file organization upon which all random-access input and output in the following programs is based. One would probably not attempt to adapt RWDISC to another system, but rather exploit the conceptual simplicity of the disc file structure and FORTRAN calls by providing similar alternatives. The first step in doing this would be to consult the report on subroutine RWDISC.[2]

(b) FRAGTAP; ORDAT: Data are assumed to originate from an RWDISC-type disc file of a particular organization used by the Moored Array Project (the "Buoy Group") at WHOI. This is referred to in the report as a TAPDIS[3] disc file. Programs FRAGTAP and ORDAT simply reorganize the data into another different type of RWDISC file to be used in all subsequent processing. These programs would probably be irrelevant to the problem of running the bispectral programs on another system.

(c) FOURIER: Besides RWDISC, the major subroutine used by this program is "HARM1", which does the Fast Fourier Transform (FFT) on the data. This is a FORTRAN-IV subroutine

which resides in the library on the system disc at WHOI.[3] Again, one could either adapt this program to another system, or provide an alternate subroutine for HARM1. It is also possible to substitute an entire program for FOURIER, in order to supply Fourier coefficients from appropriately processed data, in the order stated in the report, and in the RWDISC-type disc organization required by the bispectral programs BISCAL, BISUM, and BIVEC. There are other minor subroutines used by program FOURIER, and these will be found in Reference 3.

(d) BISCAL, BISUM, BIVEC: These are the main bispectral programs. The input is an RWDISC file of Fourier coefficients as described in the respective reports, and output is one or more consecutive disc files. The RWDISC file has been discussed above. The binary WRITE statements used to create the consecutive files should not have to be modified in another system using FORTRAN IV. The subroutines other than RWDISC used by these programs are minor and can be found in Reference 3.

(e) BPLOT, RBPLOT: These programs make contour plots from consecutive files produced by programs BISCAL, BISUM, and BIVEC. These two plotting programs use a set of standard CALCOMP-plotter subroutines which should be available on other systems.[4] These include the following:

PLOT (with entry point PLOTS)

SYMBOL

NUMBER

Additionally, programs BPLOT and RBPLOT use subroutines which reside in the system library at WHOI.[3] These include:

WHCNTR, for generating contour plots.
Has entry points: DIMWH, GRID, NOLABL, FLAGZ

AXDRAW
AXWJS
} used to draw axes via CALCOMP subroutine calls

- plus some minor programs from the WHOI library.[3]

The binary READ statements used for input should not require any change.

(f) GENRAN: This utility program to generate artificial series is intended to be run on-line in the context of the XEROX CP-V time-sharing system. It can possibly be adapted to other time-sharing systems if the required subroutines or substitutes are supplied. Program GENRAN again uses subroutine RWDISC to create a disc file organized in the RWDISC format. The only other subroutine it uses, from the WHOI system library, is subroutine NORAN, which generates pseudo-random numbers which can have a Gaussian distribution of specified mean and variance.[3]

4. References for Section IIA:

1. "Xerox Extended FORTRAN IV" Language Reference Manual, Xerox Corporation Publication No. 90/09/56E.

2. "Handbook for Computer Users," Woods Hole Oceanographic Institution, pp. V-E-1 ff.

3. "Reprints of WHOI Programs," Information Processing Center, Woods Hole Oceanographic Institution.

4. "Handbook for Computer Users," Woods Hole Oceanographic Institution, pp. V-E-36 ff.

B. BUILDING RWDISC DATA FILE PREPARATORY TO FOURIER TRANSFORM

NAMES: FRAGTAP; ORDAT

PURPOSE: To transfer data series from one or more TAPDIS[1] disc files to another RWDISC[2] random access file of different organization, for use in bispectral programs.

MACHINE: Xerox Sigma-7

SOURCE LANGUAGE: FORTRAN IV

DESCRIPTION:

Programs FRAGTAP and ORDAT are two main programs which are intended to be run consecutively in the same job. Their purpose is to create an RWDISC[2] random access file which has a different organization from the input TAPDIS file(s).[1] The major reason for doing this is that the bispectral programs which use the data tend to repeatedly access long series sequentially from the disc. The segment length into which the logical files of the new RWDISC file are divided is therefore set at 512 (the size of the monitor buffer used by RWDISC), rather than 73 in the TAPDIS file. Hence there is one disc access per 512 words rather than one per 73 as there would otherwise be. Other reasons for reorganizing the TAPDIS files are to give the user the opportunity to select from several TAPDIS files only those series of interest, and hence produce an output file which is no longer than necessary, and to simplify the structure of the RWDISC files to be input into the bispectral programs.

The reason for two main programs is to allow the opening of a newly-structured RWDISC file. Program FRAGTAP creates up to 10 consecutive files from the TAPDIS file which it reads. There is one consecutive file per TAPDIS pseudo-file[1] and an additional consecutive file for the TAPDIS label logical file. Any logical files from any TAPDIS pseudo-file may be selected for transmittal. Program FRAGTAP may be run repeatedly on several TAPDIS files to generate more consecutive files. Program ORDAT reads any or all of these consecutive files, and places the contents into logical files of the user's choice in the new RWDISC file.

INPUT: (FRAGTAP-ORDAT sequence considered as a whole) Input is one or more TAPDIS files.

OUTPUT:

I. Disc

Upon completion of the FRAGTAP-ORDAT sequence, disc output consists of a single RWDISC file having 21 logical files. The 21st logical file is reserved as a facsimile of one of the TAPDIS label files read by FRAGTAP. There is one data series per logical file in logical files 1 through 20 if all are used. It is not necessary to have 20 data series in an RWDISC file having 21 logical files. Data begins at location 5, the first three words being the Buoy-format tape* data file name from which this series originated, and the fourth word being the Buoy-format variable number.

Upon completion of the FRAGTAP program alone, there are up to ten consecutive files. One of these (normally with DCB assignment 10) is a facsimile of the TAPDIS label logical file (no. 1). The others each contain all series in sequence, called for by the user, from one TAPDIS pseudo-file, associated with one Buoy-format tape file. The data start in location 10. Locations 1 through 9 contain:

#1,2,3: Buoy tape file name associated with this pseudo-file.

#4: Number of variables read from the TAPDIS pseudo-file.

#5: Number of cycles for this TAPDIS pseudo-file (from TAPDIS label).

#6,7,8,9: Buoy format variable numbers for up to four data series.

II. Line Printer

Line printer output from a typical FRAGTAP-ORDAT sequence is shown in Appendix 1. In this job, 20 data files were transferred from a single TAPDIS file. Input and output originated and terminated on labeled tape, and the programs worked from the scratch pack.

FRAGTAP output

(1) fascimile of input data cards;

(2) starting locations of each successive pseudo-file in TAPDIS file;

(3) summary of contents of each consecutive file produced.

*This is a name used for a standard format originated and used by the Moored Array Project at Woods Hole Oceanographic Institution. See Reference 3.

ORDAT output

ORDAT furnishes a complete summary of the contents and disposition of each input consecutive file. This includes the original Buoy tape file name associated with the TAPDIS pseudo-file, the number of series transmitted from that pseudo-file, the data series length (excluding label) for that pseudo-file, the logical files (destination) of each series, and the Buoy variable numbers associated with each data series transmitted.

USAGE:

The user must first determine whether only the public (RAD) disc pack will be used, or whether he will ask for a private, scratch pack. This will be determined by the amount of data to be handled. When the scratch pack is used, labeled tape will probably also be used because of the size of the files involved. Two versions of the control card deck will be presented. The first assumes that the TAPDIS file exists on the RAD, that the final RWDISC file will be on the RAD; and that only this disc pack is used throughout. The second assumes that the RWDISC file is originally on labeled tape, that the programs will work from the scratch pack, and that the final RWDISC file will be on labeled tape.

CONTROL CARDS:

```
Let   uuu          = user number
      aaa          = account number
      INNN         = labeled tape with input TAPDIS file
      OUTT         = labeled tape with output TAPDIS file
      TAPDISFILE   = input TAPDIS file on RAD or labeled tape
      OUTFILE      = output file (21 logical files) on
                     RAD or labeled tape
```

Suppose 20 series from 9 TAPDIS pseudo-files from a single TAPDIS file are to be placed in logical files 1 through 20 in order, in the output file.

Version 1 (Everything on RAD)

```
                  !JOB aaa,uuu
                  !LIMIT (TIME,1),(CORE,26)
                  !ASSIGN F:RAD,(FILE,TAPDISFILE),(DIRECT),(KEYED)
                  !ASSIGN F:1,(FILE,T1),(OUT),(SAVE)
                  !ASSIGN F:2,(FILE,T2),(OUT),(SAVE)
                            : (consecutive files 3 through 9)
(for label file)!ASSIGN F:10,(FILE,T10),(OUT),(SAVE)
                  !LOAD (EF,(FRAGTAPR)),(UNSAT,(3))
                  !RUN
                  !DATA
```

```
———————→ FRAGTAP data cards
!ASSIGN F:RAD,(FILE,OUTFILE),(DIRECT),(KEYED),(OUTIN),(SAVE)
!ASSIGN F:1,(FILE,T1)
         ⋮ (Consecutive files 2 through 9)
!ASSIGN F:10,(FILE,T10)
!LOAD (EF,(ORDATR)),(UNSAT,(3))
!RUN
!DATA
———————→ ORDAT data cards
```

Version 2 (Initial input, final output on labeled tape; all work done from scratch pack)

```
!JOB aaa,uuu
!LIMIT (TIME,1),(CORE,26),(9T,1),(SP,1),(MOUNT,(X,PACK))
!MESSAGE USES 9T INNN
!MESSAGE USES 9T OUTT***WRITE
!SPAK (NRAN,1),(1500)
!ASSIGN M:EO,(INOUT)
!PCL
COPY LT#INNN/TAPDISFILE TO DP#PACK/R1.:PAK(WR(ALL),FA)
REM LT#INNN
END
!ASSIGN F:RAD,(SN,PACK),(FILE,R1,:PAK),(INOUT)
!ASSIGN F:1,(SN,PACK),(FILE,C1,:PAK),(INOUT)
             ⋮ (Consecutive files 2 through 9)
!ASSIGN F:10,(SN,PACK),(FILE,C10,:PAK),(INOUT)
!LOAD (EF,(FRAGTAPR)),(UNSAT,(3))
!RUN
!DATA
———————→ FRAGTAP data cards
!ASSIGN M:EO,(OUT)
!LOAD (EF,(ORDATR((,(UNSAT,(3))
!RUN
!DATA
———————→ ORDAT data cards
!PCL
SPE LT#OUTT
COPY DP#PACK/R1.:PAK TO LT#OUTT/OUTFILE(WR(ALL),FA)
END
```

Note in the above that program FRAGTAP could have been run more than once to access several TAPDIS files, and that program ORDAT need not use all consecutive files produced by FRAGTAP. There cannot be more than 10 DCB assignments at any time, however.

DATA CARDS:

All data card entries are in FORTRAN IV free-field generalized format.

FRAGTAP

Card(1): NFILES- number of pseudo-files (one per Buoy-format tape file) to access from TAPDIS file

JDCB - NFILES DCB assignments (one consecutive file per TAPDIS pseudo-file)

LDCB - one DCB assignment for TAPDIS label logical file

Cards(2 through NFILES+1)

NSEQ - sequential position of TAPDIS pseudo-file

NV - number of variables to access from each TAPDIS pseudo-file

ILF - actual RWDISC logical files of the NV variables in TAPDIS pseudo-file

ORDAT

Card(1): NDCB - number of consecutive files, not counting label file

LDCB - DCB for label file

Cards(2 through NDCB+1)

IDCB - DCB of a consecutive file to read (applies to this entire input card)

NSERIES - number of data series in consecutive file having IDCB DCB assignment

ILF - NSERIES logical files in output RWDISC file, in which to place data series from consecutive file associated with IDCB

RESTRICTIONS:

(1) No more than ten DCB assignments for consecutive files can be in effect at any time.

(2) No logical file past the 20th can be used for data in the output RWDISC file. The 21st file is reserved for labels.

(3) The limiting data series length is 16000 words.

STORAGE REQUIREMENTS:

Peak core cited in a typical run of FRAGTAP-ORDAT was 50 512-word pages.

SUBPROGRAMS REQUIRED: None

TIMING: The processing of 20 12000-word data series on the scratch pack, starting and ending with RWDISC files on labeled tape, took 0.735 min of CPU time.

ERRORS AND DIAGNOSTICS:

I. FRAGTAP

(1) NERR = ...followed by "STOP 2"
RDISC[2] error (type NERR) in reading word 100 (number of pseudo-files) in TAPDIS label logical file.

(2) NERR = ...followed by "STOP 3"
RDISC error (type NERR) in reading total TAPDIS label logical file.

(3) NERR = ...followed by "STOP 4"
RDISC error (type NERR) in reading a TAPDIS data file.

(4) ISTAT = ...followed by "STOP 3"
BUFFER OUT error (type ISTAT) in writing labels to a consecutive data file.

(5) ISTAT = ...followed by "STOP 1"
BUFFER OUT error (type ISTAT) in writing fascimile of TAPDIS label file to consecutive file.

(6) ISTAT = ...followed by "STOP 4"
BUFFER OUT error (type ISTAT) in writing a consecutive file of data.

II. ORDAT

(1) ISTAT = ...followed by "STOP 1"
BUFFER IN error (type ISTAT) in reading label portion of a consecutive file.

(2) INWDS = ...followed by IDCB = ...followed by "STOP 10"
BUFFER IN signalled something other than 9 words for number read in label portion of consecutive file with IDCB

(3) IDCB = ...; NUMVAR = ...; NSERIES = ...;
"STOP DISCREPANCY IN NO. OF VARIABLES"
Number of series specified by user (NSERIES) disagrees with that in TAPDIS label file for pseudo-file corresponding to IDCB.

(4) N = ...; INWDS=...;JCYC = ...; "STOP 50"
No. of words of data(INWDS) read by BUFFER IN disagrees with that in TAPDIS label file.

(5) ISTAT = ...; "STOP 2"
BUFFER IN error (type ISTAT) in reading consecutive file containing TAPDIS labels.

PROGRAMMER: GERARD H. MARTINEAU

ORIGINATOR: MELBOURNE G. BRISCOE

DATE: June, 1977

REFERENCES:

1. Report on Program TAPDIS by John Maltais, Woods Hole Oceanographic Institution.

2. "Handbook for Computer Users," Information Processing Center, Woods Hole Oceanographic Institution, pp. V-E-1 ff.

3. "A Nine Channel Digital Magnetic Tape Format for Storing Oceanographic Data," by John A. Maltais. Woods Hole Oceanographic Institution Reference No. 69-55.

APPENDIX 1
(FRAGTAP; ORDAT)

```
 13:29 MAR 25,'77 ID=0377
JOB 462,1719
LIMIT (TIME,4),(CORE,30),(9T,1),(SP,1),(MOUNT,(X,PACK))
MESSAGE USES 9T IWJM
MESSAGE USES 9T IWJN***WRITE
SPAK (NRAN,1),1500
ASSIGN M:EO,(INPUT)
PCL
COPY LT#IWJM/IWEX   TO DP#PACK/R1.:PAK(WR(ALL),FA)
REM LT#IWJM
END
PCL PROCESSING TERMINATED
ASSIGN F:RAD,(SN,PACK),(FILE,R1,:PAK),(INPUT)
ASSIGN F:1,(SN,PACK),(FILE,C1,:PAK),(INPUT)
ASSIGN F:2,(SN,PACK),(FILE,C2,:PAK),(INPUT)
ASSIGN F:3,(SN,PACK),(FILE,C3,:PAK),(INPUT)
ASSIGN F:4,(SN,PACK),(FILE,C4,:PAK),(INPUT)
ASSIGN F:5,(SN,PACK),(FILE,C5,:PAK),(INPUT)
ASSIGN F:6,(SN,PACK),(FILE,C6,:PAK),(INPUT)
ASSIGN F:7,(SN,PACK),(FILE,C7,:PAK),(INPUT)
ASSIGN F:8,(SN,PACK),(FILE,C8,:PAK),(INPUT)
ASSIGN F:9,(SN,PACK),(FILE,C9,:PAK),(INPUT)
ASSIGN F:10,(SN,PACK),(FILE,C10,:PAK),(INPUT)
LOAD (EF,(FRAGTAPR)),(UNSAT,(3))
:P1 ASSOCIATED.


* * ALLOCATION SUMMARY * *

  PROTECTION        LOCATION    PAGES

DATA (00)             A000        27
PROCEDURE (01)        F400         2
DCB (10)              EE00         3
********************************    SGN= C7L                       SIZE=020.9K     *****

 ******** PROTECTION TYPES:  00 DATA          01 PROCEDURE    10 STATIC

                             SEGHI-0  ED91   SEGHI-1  F735   SEGHI-2  F3FF
                             SEGLO-0  A000   SEGLO-1  F400   SEGLO-2  EE00

                             00 SIZE= 4D92   01 SIZE=  336   10 SIZE=  600
RUN
INPUT RECORDS FOLLOW (IN GENERALIZED FORMAT)
9 1 2 3 4 5 6 7 8 9 10
 1 2 2 3
 2 3 2 3 4
 3 1 2
 4 3 2 3 4
 5 1 2
 6 2 2 3
 7 2 2 3
 8 3 2 3 4
 9 3 2 3 4

   SEQUENTIAL         REL LOCATION
 INPUT FILE READ       IN TAPDIS
    BY TAPDIS         FILE (KREL)
        1                   1
```

```
        2                     12001
        3                     24001
        4                     36001
        5                     48001
        6                     60001
        7                     72001
        8                     84001
        9                     96001

    OUTPUT FILE SUMMARY IN ORDER OF ACCESS

 OUTPUT                            BUOY TAPE VAR'S        TOTAL WORDS
  DCB       TAPE FILE            ACCESSED THIS FILE       IN OUTPUT FILE

   1        IWEX A1     3         2    3                   12009
   2        IWEX A2               1    2    3              12009
   3        IWEX B2     3         3                        12009
   4        IWEX A4               1    2    3              12009
   5        IWEX A5               1                        12009
   6        IWEX A6               1    2                   12009
   7        IWEX B6               1    3                   12009
   8        IWEX A8               1    2    3              12009
   9        IWEX C10    3         1    2    3              12009
 *STOP* OUTPUT COMPLETED
```

```
ASSIGN M:EO,(OUT)
LOAD (EF,(ORDATR)),(UNSAT,(3))
!P1 ASSOCIATED.


* * ALLOCATION SUMMARY * *

  PROTECTION      LOCATION    PAGES

DATA (00)           A000       27
PROCEDURE (01)      F400        2
DCB (10)            EE00        3
******************************    SGN= C7L                      SIZE=020.5K    *****

 ******** PROTECTION TYPES:  00 DATA          01 PROCEDURE    10 STATIC

                             SEGHI=0   ECB3   SEGHI=1   F6A1   SEGHI=2   F3FF
                             SEGLO=0   A000   SEGLO=1   F400   SEGLO=2   EE00

                             00 SIZE= 4CB4    01 SIZE=  2A2    10 SIZE=   600
RUN

OUTPUT SUMMARY FOR DCB  1
               BUOY FILE:  IWEX A1     3
           NO. OF SERIES: 2
   DATA WDS. EACH SERIES: 12000
   WRITTEN TO LOG. FILES:   1   2
          BUOY VAR. NOS.:  2   3


OUTPUT SUMMARY FOR DCB  2
               BUOY FILE:  IWEX A2
           NO. OF SERIES: 3
   DATA WDS. EACH SERIES: 12000
   WRITTEN TO LOG. FILES:   3   4   5
```

```
               BUOY VAR. NOS.:  1    2    3


OUTPUT SUMMARY FOR DCB  3
                BUOY FILE:  IWEX B2    3
          NO. OF SERIES:  1
   DATA WDS. EACH SERIES:  12000
   WRITTEN TO LOG. FILES:    6
          BUOY VAR. NOS.:  3


OUTPUT SUMMARY FOR DCB  4
                BUOY FILE:  IWEX A4
          NO. OF SERIES:  3
   DATA WDS. EACH SERIES:  12000
   WRITTEN TO LOG. FILES:    7    8    9
          BUOY VAR. NOS.:  1    2    3


OUTPUT SUMMARY FOR DCB  5
                BUOY FILE:  IWEX A5
          NO. OF SERIES:  1
   DATA WDS. EACH SERIES:  12000
   WRITTEN TO LOG. FILES:   10
          BUOY VAR. NOS.:  1


OUTPUT SUMMARY FOR DCB  6
                BUOY FILE:  IWEX A6
          NO. OF SERIES:  2
   DATA WDS. EACH SERIES:  12000
   WRITTEN TO LOG. FILES:   11   12
          BUOY VAR. NOS.:  1    2


OUTPUT SUMMARY FOR DCB  7
                BUOY FILE:  IWEX B6
          NO. OF SERIES:  2
   DATA WDS. EACH SERIES:  12000
   WRITTEN TO LOG. FILES:   13   14
          BUOY VAR. NOS.:  1    3


OUTPUT SUMMARY FOR DCB  8
                BUOY FILE:  IWEX A8
          NO. OF SERIES:  3
   DATA WDS. EACH SERIES:  12000
   WRITTEN TO LOG. FILES:   15   16   17
          BUOY VAR. NOS.:  1    2    3


OUTPUT SUMMARY FOR DCB  9
                BUOY FILE:  IWEX C10    3
```

```
                BUBY VAR. NBS.:  1    2    3


OUTPUT SUMMARY FBR DCB  3
                   BUBY FILE:  IWEX B2     3
              NB. BF SERIES: 1
    DATA WDS. EACH SERIES: 12000
    WRITTEN TB LBG. FILES:   6
                BUBY VAR. NBS.:  3


OUTPUT SUMMARY FBR DCB  4
                   BUBY FILE:  IWEX A4
              NB. BF SERIES: 3
    DATA WDS. EACH SERIES: 12000
    WRITTEN TB LBG. FILES:   7    8    9
                BUBY VAR. NBS.:  1    2    3


OUTPUT SUMMARY FBR DCB  5
                   BUBY FILE:  IWEX A5
              NB. BF SERIES: 1
    DATA WDS. EACH SERIES: 12000
    WRITTEN TB LBG. FILES:  10
                BUBY VAR. NBS.:  1


OUTPUT SUMMARY FBR DCB  6
                   BUBY FILE:  IWEX A6
              NB. BF SERIES: 2
    DATA WDS. EACH SERIES: 12000
    WRITTEN TB LBG. FILES:  11  12
                BUBY VAR. NBS.:  1    2


OUTPUT SUMMARY FBR DCB  7
                   BUBY FILE:  IWEX B6
              NB. BF SERIES: 2
    DATA WDS. EACH SERIES: 12000
    WRITTEN TB LBG. FILES:  13  14
                BUBY VAR. NBS.:  1    3


OUTPUT SUMMARY FBR DCB  8
                   BUBY FILE:  IWEX A8
              NB. BF SERIES: 3
    DATA WDS. EACH SERIES: 12000
    WRITTEN TB LBG. FILES:  15   16   17
                BUBY VAR. NBS.:  1    2    3


OUTPUT SUMMARY FBR DCB  9
                   BUBY FILE:  IWEX C10     3
```

```
          NO. OF SERIES: 3
DATA WDS. EACH SERIES: 12000
WRITTEN TO LOG. FILES:  18   19   20
       BUOY VAR. NOS.:   1    2    3


 900 WORDS WRITTEN IN LABEL L. F. NO. 21
*STOP* NORMAL END
PCL
SPE LT#TWJN
COPY DP#PACK/R1.:PAK TO LT#TWJN/TWEX22(WR(ALL),FA)
END
PCL PROCESSING TERMINATED
```

C. MODIFYING RWDISC FILE FROM B BY REPLACING DATA SERIES WITH FOURIER COEFFICIENTS IN A SPECIAL ORDER

NAME: FOURIER

TYPE: Main Program

PURPOSE: To create RWDISC[1] file of FOURIER coefficients from RWDISC file of data series, for input into bispectral programs

MACHINE: SIGMA-7

SOURCE LANGUAGE: FORTRAN IV

DESCRIPTION:

Program FOURIER takes an RWDISC file containing a number of data series and replaces each one by a series of Fourier coefficients which is specially formatted for efficient access by the bispectral programs written by this author. The series may be processed in a number of ways including: a) prewhitening (done on the original series before any other processing); b) compression of the series by triangular weighting on consecutive subsamples (done after any prewhitening but before any further operations); c) splitting into pieces, which may be contiguous or: d) overlapping of pieces by 50%; e) subtraction of mean of each piece, or trend based on endpoints, or trend based on linear regression; f) "Hanning" of pieces, i.e., 100 percent cosine tapering.

Calculation of the Fourier coefficients is done by IPC program HARM1.[2]

INPUT:

Input is assumed to be an RWDISC file consisting of one data series per logical file, starting at word 5 of each logical file, the first four words being reserved for label information. Every series to be accessed by program FOURIER is assumed to be of equal length. There may be up to 21 logical files in the input RWDISC file. If there are 1 through 20 inclusive, every logical file may contain a data series. The case of 21 logical files is considered to be standard, and in that case the 21st logical file is reserved for label information.

<u>OUTPUT:</u> (1) Disc

Disc output is a modified version of the input RWDISC file, and therefore has the same number of logical files. Each data series is replaced by a series of Fourier coefficients starting at word 5 and organized as follows:

Let the n-point data series be represented by the sequence:

$$\{Y_j\},\ j = 0,\ n-1$$

and the Y_j-th term by the expression:

$$Y_j = a_0 + \sum_{k=1}^{n/2} [a_k \cos(2\pi jk/n) + b_k \sin(2\pi jk/n)] \ .$$

Now let a_{km}, b_{km} represent the k-th order coefficients, for the m-th piece. Further, let $\{a_{km}\}$ represent the sequence of k-th order cosine coefficients over all pieces taken in order, the overlapped pieces contributing the even sequence m = (2,4,6,...) in the case of overlapping. Then a logical file of Fourier coefficients produced by program FOURIER has the appearance

$$\{a_{0m}\},\ [\{\{a_{km}\},\{b_{km}\}\},\ k = 1,\ n/2-1],\ \{a_{n/2\ m}\}$$

All coefficients $b_{n/2\ m}$ are zero in the above representation of the data series.

It will be noted that the coefficients necessary to calculate a spectral estimate at a given frequency are always contiguous.

(2) Line Printer (see Appendix 2)

Normally, line printer output consists of three lines giving the processing conditions of this run, and the date of the run. Line 3 is a facsimile of an EBCDIC label written on logical file 21 of the output RWDISC file (starting at word 1) if 21 logical files are chosen. The entries are self-explanatory if it is noted that when referring to the execution of a given process, 0 means "no" and 1 means "yes". The entry "NO WDS DATA" refers to the length of each data series entering the program.

USAGE:

(1) Control cards

Let aaa = account number

uuu = user number

fff = input name

and let all element files exist in account 462. Then the control card portion of the input card deck appears as follows:

```
  !JOB aaa,uuu
  !LIMIT (TIME,4),(CORE,50)
  !ASSIGN F:RAD,(FILE,fff),(DIRECT),(KEYED),(INOUT)
* !LOAD(EF,(FOURIERR,462),(CSUBR,462),(CTRLR,462),(INPUTR,462),;
  !(KOEFHARMR,462),(OUTOMEGAR,462)),(UNSAT,(3))
  !RUN
  !DATA
```

* If all element files are contained in a single file FOURIERR, then these two cards become simply: !LOAD (EF,(FOURIERR,462)),(UNSAT,(103),(3))

(2) Data cards

There is a single data card followed by NAMELIST input, which must be terminated by an * card, whether or not there is any actual namelist input.

The data card has 8 or 9 entries, all in FORTRAN generalized format. The order of these entries is:

IFIRST - the first logical file to be processed by program FOURIER

ILAST - the last logical file to be processed by program FOURIER. See "RESTRICTIONS"

NDO2 - the data length in words

IPREW - prewhitening switch (0 or 1)

ISUBSAMP - subsampling switch (0 or 1). If ISUBSAMP = 1, triangular weighting of the subsample will be done automatically.

NSSMP - an <u>odd number</u>, the length of each subsample; to be entered <u>only</u> if ISUBSAMP = 1

LPIECE - piece length. <u>SEE RESTRICTIONS BELOW</u>

IOVERLAP - 50% overlap switch (0 or 1). This will be set to 0 unless Hanning is specified

ISUBT - 0 Don't subtract mean or trend of each piece
1 Subtract trend based on endpoints of each piece
2 Subtract trend based on linear regression
3 Subtract mean of each piece

IHANN - Hanning switch (0 or 1)

The NAME LIST at present has only one relevant variable: MF = no. of logical files in the RWDISC file. The default for MF is 21, and it is absolutely essential to enter it as a name list variable if MF is not in fact 21. Failure to do so will not produce an error message, and can result in subtly incorrect answers.

If MF is four, for example, follow the data card by

MF = 4 .

If MF is twenty-one, follow the data card simply by

* .

RESTRICTIONS:

(1) NDO2 $\leq$ 16000. If NPIECES is the total number of pieces, then in the limit as NPIECES $\rightarrow \infty$, the total length of the output series of Fourier coefficients will approach 2 $\times$ NDO2. If the user wishes to process many series of length considerably shorter than 16000, it may be advisable to recompile the program. The main working array WK is in COMMON and is normally dimensioned WK(33000). When editing the source program, this array dimension should be set at (2$\times$NDO2). This is most readily done in the COMMON of every program and subroutine by combining all into a single file.

(2) LPIECE. There are several important restrictions on the length of a piece; all of which must be met before the program will proceed:

i) LPIECE must be an even number with no prime factor greater than 5.[2] Additionally, it is assumed that the output of program FOURIER will be used for bispectral calculations. The scalar bispectral program BISCAL requires that LPIECE be evenly divisible by 4, and the vector (rotary) bispectral program

BIVEC requires that LPIECE be evenly divisible by 8. Hence these added restrictions are placed on LPIECE. Appendix 1 is a list of piece lengths $\leq$ 4000 which can be used for programs BISCAL and BIVEC. Numbers in parentheses are not divisible by 8, and so cannot be used in connection with program BIVEC. Piece lengths must further meet the following criteria:

ii) LPIECE must satisfy one of the following inequalities: Let IND be the data series length after any subsampling. Then:

$$450 \geq \text{LPIECE} \geq (\text{IND}/75) \text{ for 50\% overlap}$$

- or -

$$450 \geq \text{LPIECE} \geq (\text{IND}/150 \text{ for no overlap}$$

These limits can be changed by minor source program modifications. Briefly, to increase the upper limit, change the dimension of all arrays of size 450 to the new upper limit. To decrease the lower limit, all arrays of size 150 must be changed to size J, where the denominator of the lower limit above is (J/2) for 50% overlap and J for no overlap.

(3) i) MF $\leq$ 21

ii) If MF = 21, then ILAST < 21

iii) If MF < 21, then ILAST $\leq$ MF

<u>ERRORS AND DIAGNOSTICS:</u>

(1) "PIECE LENGTH OF...INVALID. MUST BE FROM THE FOLLOWING SET:"

Meaning: The user has specified a piece length which fails to meet criterion i), item (2) of "RESTRICTIONS" section for program BISCAL. The piece lengths which are output are those of Appendix 1.

Action: Abort

(2) "PIECE LENGTH DOES NOT SATISFY RESTRICTION FOR (NO) OVERLAP"

Meaning: The user has specified a piece length which does not satisfy one of the criteria ii), item (2) of "RESTRICTIONS"

Action: Abort

(3) "SUBSAMPLE LENGTH MUST BE ODD"

Meaning: The user has specified subsampling with an even subsample length.

Action: Abort

(4) "OVERLAP POSSIBLE ONLY IF HANN. IOVERLAP SET TO 0"

Meaning: Self-explanatory

Action: Proceed with no overlapping.

(5) "MF CANNOT EXCEED 21"

Meaning: User must not specify more than 21 logical files

Action: Abort

(6) "LAST LOG FILE CANNOT EXCEED MF OR 20"

Meaning: One of the following conditions was not met:

(a) If MF = 21, the last logical file to be processed must be less than 21;

(b) If MF < 21, the last logical file must not exceed MF.

Action: Abort

PROGRAMMER: GERARD H. MARTINEAU

ORIGINATOR: MELBOURNE G. BRISCOE

DATE: June, 1977

REFERENCES: (1) "Handbook for Computer Users," Woods Hole Oceanographic Institution, pp V-E-1 ff.

(2) "Reprints of WHOI Programs," Woods Hole Oceanographic Institution, HARMI Report.

APPENDIX 1
(FOURIER)

APPENDIX 1

Numbers divisible by four and ≤ 4000 containing no prime factor greater than 5:

(Numbers in parentheses may not be used for piece lengths to be handled by program BIVEC)

(12)	(108)	(324)	768	(1500)	2560
16	120	360	800	1536	2592
(20)	128	384	864	1600	(2700)
24	144	400	(900)	(1620)	2880
32	160	432	960	1728	2916
(36)	(180)	480	(972)	1800	3000
40	(192)	(500)	(1000)	1920	3072
48	200	512	1024	1944	3200
(60)	216	(540)	1080	2000	3240
64	240	576	1152	2048	3456
72	256	600	1200	2160	3600
80	288	640	1280	2304	3848
96	(300)	648	1296	2400	3888
(100)	320	720	1440	(2500)	4000

APPENDIX 2
(FOURIER)

```
********************************** SIGMA 5/6/7/9  LOAD MODULE MAP  **********************************
********************************* ACN= 462       <ROOT> START=1540C  *********************************
******************************** LMN= D7L              BIAS= A000   ********************************
******************************* SGN= D7L              SIZE=012.9K  ********************************
                + =DBLE DEF  - =LIB DEF  * =UNUSED DEF  @ =UNUSED LIB DEF

******** PROTECTION TYPES:  00 DATA        01 PROCEDURE    10 STATIC

                            SEGHI-0 1507D  SEGHI-1 1584F  SEGHI-2 153FF
                            SEGLO-0  A000  SEGLO-1 15400  SEGLO-2 15200

                            00 SIZE= B07E  01 SIZE=  750  10 SIZE=  200

*******  SECT - PROGRAM SECTIONS MAP             ******
 A2F8 DSECT 0   A2F8 0  F4:COM                                                         - 82E6 WDS
12642 CSECT 0  12642 0 *V#4MAIN                                                        -  3D4 WDS
12A16 CSECT 0  12A16 0 *V#CTRL                                                         -  3DC WDS
12DE6 CSECT 0  12DE6 0 *V#INPUT                                                        -  3DE WDS
131C4 CSECT 0  131C4 0 *V#KOEFF                                                        -   16 WDS
131DA CSECT 0  131DA 0 *V#OUTOMEGA                                                     - 139A WDS
1521E LDCB  2  1521E 0  M:*                                                            -   2A WDS
15250 LDCB  2  15250 0  M:DO                                                           -   34 WDS
1528C LDCB  2  1528C 0  M:OC                                                           -   34 WDS
152C8 LDCB  2  152C8 0  M:LL                                                           -   34 WDS
15304 LDCB  2  15304 0  F:108                                                          -   34 WDS
15340 LDCB  2  15340 0  F:105                                                          -   34 WDS
1537C LDCB  2  1537C 0  F:RAD                                                          -   34 WDS
1540C CSECT 1  1540C 0 *4MAIN                                                          -   92 WDS
1549E CSECT 1  1549E 0  CTRL                                                           -   64 WDS
15502 CSECT 1  15502 0  INPUT                                                          -   F8 WDS
155FA CSECT 1  155FA 0  KOEFF                                                          -   FE WDS
156F8 CSECT 1  156F8 0  OUTOMEGA                                                       -   70 WDS
15A24 CSECT 1                                                                          -  12A WDS

*******  RELOCATABLE DEFINITIONS SORTED BY NAME  ******
           1549E 0  CTRL        1537C 0  F:RAD        15340 0  F:105     15304 0  F:108
            A2F8 0  F4:COM      15502 0  INPUT        155FA 0  KOEFF     15433 0 *LABEL
           1521E 0  M:*         15250 0  M:DO         152C8 0  M:LL      1528C 0  M:OC
           156F8 0  OUTOMEGA    155E5 0 *PREWHITE     12A16 0 *V#CTRL    12DE6 0 *V#INPUT
           131C4 0 *V#KOEFF     131DA 0 *V#OUTOMEGA   12642 0 *V#4MAIN   1540C 0 *4MAIN

RUN

LABEL AS WRITTEN ON L.F. 21 IF MF=21:
 NO LF'S  4*NO WDS DATA  9728*PREW 0*SUBSAMP 0*NSUBSAMP   0*PC SIZE   256*NO PCS  38*OLAP 1*ISUBT 3*HANN 0 *CREATED 20:43 MAY 19,'77

*STOP* PROCESSING COMPLETED
```

D. BISPECTRAL CALCULATIONS AND PLOTTING

NAME: BISCAL

TYPE: Main Program

PURPOSE: To calculate auto and cross-bispectra of two real series, to write a consecutive file of bicoherences for transmittal to plotting program BPLOT,[1] and to compute confidence levels for the bicoherences.

MACHINE: Xerox Sigma-7

SOURCE LANGUAGE: FORTRAN IV

DESCRIPTION:

Program BISCAL uses as input an RWDISC[2] file of FOURIER coefficients having a particular order to be described below. These coefficients are assumed to result from real data series and are normally the output of program FOURIER,[3] which has done any necessary prewhitening, subsampling, overlapping, mean and trend removal, and Hanning. If L is the piece length and NP is the number of pieces resulting from program FOURIER, there are (L)(NP) words input into BISCAL for each data series.

Program BISCAL then computes approximately $(3/4)(L/2)^2$ values of bicoherence, along with the corresponding biphase and biphase error if desired. Two alternate forms of bicoherence may be calculated. The default form is that defined by N. C. G. Yao.[4] The other form is a normalization by amplitude, so that only phase information remains (see below). The user can elect to have the program write a consecutive file of bicoherences to be used as input to program BPLOT for plotting. The user can also instruct the program to determine those bicoherences which correspond to specified confidence levels. A listing of partitioning of bicoherences into bins of specified size can be obtained, as well as a listing of bicoherences and related quantities when the bicoherences fall between limits set by the user.

Computations Performed

A. Bispectra

Form 1

The default definitions of bispectrum, bicoherence and biphase used by program BISCAL are those given by N. C. G. Yao[4] except for the definition of the "sum frequency" ω_3. Consider two real time series from which are derived three complex Fourier coefficients, each of which may come from either series. Let these coefficients be $X(\omega_1)$, $X(\omega_2)$, and $X(\omega_3)$. Since we are considering real time series, the real part of $X(\omega_k)$ is an even function and the imaginary part is an odd function about the frequency origin. Using this property,

$$X(\omega_k) = \begin{cases} a_k + ib_k & \text{when} \quad \omega_k > 0 \\ \\ a_k - ib_k & \text{when} \quad \omega_k < 0 \, . \end{cases} \tag{1}$$

The average bispectrum over the pieces under consideration is then defined as $\langle B(\omega_1,\omega_2)\rangle$, where:

$$\begin{aligned} \langle B(\omega_1,\omega_2)\rangle &= \langle X(\omega_1)X(\omega_2)X^*(\omega_3)\rangle \qquad && \text{when } \omega_1 + \omega_2 = \omega_3 \\ &= 0 && \text{when } \omega_1 + \omega_2 \neq \omega_3 \end{aligned} \tag{2}$$

In the frequency domain of interest (see below), $\omega_1 > 0$ always, and $|\omega_1| > |\omega_2|$. Since $\omega_3 = \omega_1 + \omega_2$, we have that $\omega_3 > 0$ in this domain always, and therefore

$$X^*(\omega_3) = a_3 - ib_3$$

The second Fourier coefficient $X(\omega_2)$ ranges over both positive and negative frequencies in the domain of interest and is evaluated according to equation (1).

Expanding equation (2) above, we have that:

$$\mathrm{Re}\langle B(\omega_1,\omega_2)\rangle = \langle a_1a_2a_3 \pm a_1b_2b_3 + b_1a_2b_3 \mp b_1b_2a_3\rangle \qquad (2a)$$

$$\mathrm{Im}\langle B(\omega_1,\omega_2)\rangle = \langle \pm a_1b_2a_3 - a_1a_2b_3 + b_1a_2a_3 \pm b_1b_2b_3\rangle \qquad (2b)$$

where the upper signs correspond to $\omega_2 > 0$ and the lower to $\omega_2 < 0$.

The bicoherence bic (ω_1,ω_2) is then defined as:

$$\mathrm{bic}(\omega_1,\omega_2) = \frac{|\langle B(\omega_1,\omega_2)\rangle|}{[\langle|X(\omega_1)|^2\rangle\langle|X(\omega_2)|^2\rangle\langle|X(\omega_1+\omega_2)|^2\rangle]^{1/2}} \qquad (3)$$

and the biphase [biphase (ω_1,ω_2)] as:

$$\mathrm{biphase}(\omega_1,\omega_2) = \tan^{-1}\frac{\mathrm{Im}\langle B(\omega_1,\omega_2)\rangle}{\mathrm{Re}\langle B(\omega_1,\omega_2)\rangle} \qquad (4)$$

Form 2

Program BISCAL can compute an alternate form of bicoherence based on an amplitude-normalized bispectrum. Let:

$$B_R = \mathrm{Real}[X(\omega_1)X(\omega_2)X^*(\omega_1+\omega_2)]$$

$$B_I = \mathrm{Imag}[X(\omega_1)X(\omega_2)X^*(\omega_1+\omega_2)]$$

$$B = \sqrt{B_R^{\,2} + B_I^{\,2}}$$

The average bispectrum over the pieces under consideration is then defined as $\langle B'(\omega_1,\omega_2)\rangle$, where:

$$\langle B'(\omega_1,\omega_2)\rangle = \left\langle \frac{B_R}{B} + i\frac{B_I}{B} \right\rangle = \left\langle \frac{B_R}{B} \right\rangle + i\left\langle \frac{B_I}{B} \right\rangle .$$

The amplitude-normalized bicoherence is then defined as:

$$\mathrm{bic}'(\omega_1,\omega_2) = |\langle B'(\omega_1,\omega_2)\rangle|$$

The biphase is now:

$$\mathrm{biphase}(\omega_1,\omega_2) = \tan^{-1} \frac{\mathrm{Im}\langle B'(\omega_1,\omega_2)\rangle}{\mathrm{Re}\langle B'(\omega_1,\omega_2)\rangle}$$

<u>Frequency Domains</u>

Because of symmetries in the bispectrum, it is not necessary to let the constituent frequencies range over all possible values. If all three coefficients in (1) above are based on the same series, then the "auto-bispectrum" and "auto-bicoherence" are being calculated and the minimum necessary domain is:

$$\{A\} = \{\{\omega_2 \leq \omega_1\} \cap \{\omega_2 \leq \omega_1 + L/2\} \cap \{\omega_2 > 0\}\} \tag{5}$$

where $\{\omega_2 \leq \omega_1\}$ actually means $\{(\omega_1,\omega_2) \ni \omega_2 \leq \omega_1\}$ etc., and where $\omega_1 + \omega_2 - \omega_3 = 0$ and L is the piece length, or pictorially:

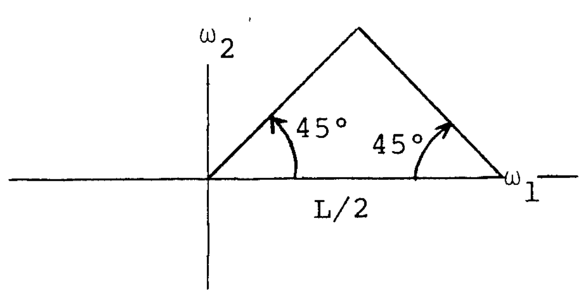


Figure 1

Whenever more than one data series enter the computation of the three coefficients in (1), then the minimum necessary domain becomes:

$$\{C\} = \{\{A\} \cup \{\{\omega_2 > -\omega_1\} \cap \{\omega_1 \leq L/2\} \cap \{\omega_2 < 0\}\}\} \tag{6}$$

where $\omega_1 + \omega_2 - \omega_3 = 0$, L is the piece length, and {A} is defined in (5), or pictorially:

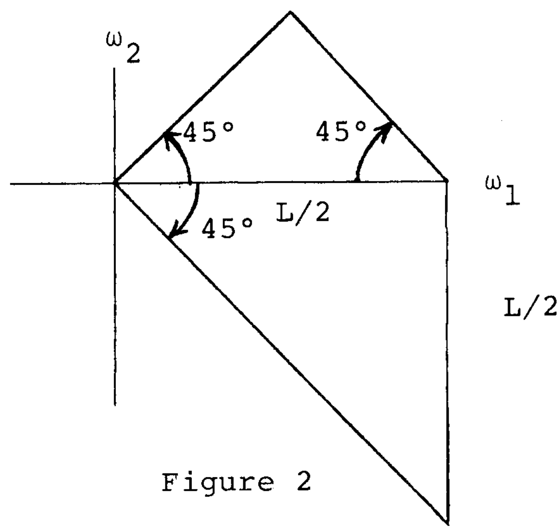


Figure 2

In this case, we say that "cross-bispectrum" and "cross-bicoherence" are being computed.

<u>Order of the Computations</u>

Program BISCAL can handle two data series, each of length 16K words. This means that for 50% overlap and for the limiting case of an infinite number of pieces, each data series entering the program consists of 32K FOURIER coefficients. In order to permit computation of the bispectra, and to minimize the number of disc accesses, the following steps are taken:

1) Organization of input file

   Let $\{a_i\}$ be the set of all real parts of the i^{th}-frequency FOURIER coefficient taken over all pieces, and let $\{b_i\}$ be the corresponding imaginary parts. Then, for a piece length of L, the input file elements have the following order:

$\{a_0\}, \{a_1\}, \{b_1\}, \{a_2\}, \{b_2\}, \ldots \{a_{\frac{L}{2}-1}\}, \{b_{\frac{L}{2}-1}\}, \{a_{\frac{L}{2}}\}$

The elements of $b_{\frac{L}{2}}$ are all zero and do not appear in the input file. Program FOURIER[3] automatically outputs coefficients in this order. This permits all elements associated with a given frequency to be input by entering the series only once.

2) Organization of frequency domain and computation procedure

No more than (L/2 + 2)(NP) elements from each series are in core at any time, where L = piece length, and NP = no. of pieces. These elements correspond essentially to the "top" and "bottom" halves of each series. This causes the domain of Fig. 2 to divide naturally into five sub-domains or "rasters" as indicated by the dashed lines in the following figure:

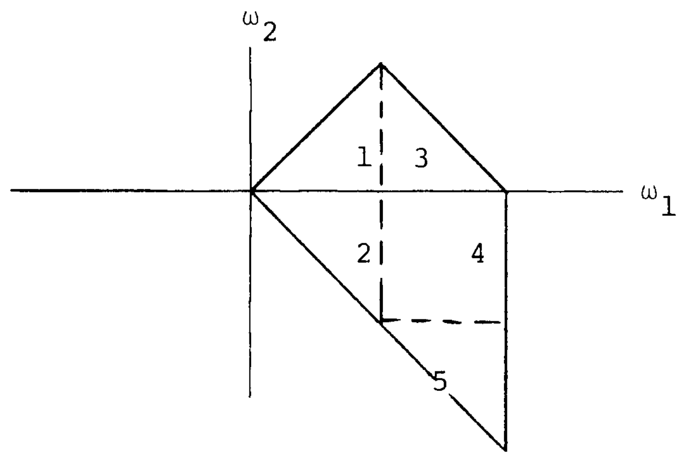


Figure 3

Computations of bicoherences proceed for each raster in order. The reading of series "halves" from the disc is a function of each particular combination of source series for the three frequencies, an attempt being made to maximize efficiency. Within each raster, computation of bicoherences is done from left to right along diagonal scans of slope (-1), each scan thus corresponding to a single sum frequency ω_3. The disc is accessed for the coefficients of frequency ω_3 only when these coefficients are not already available in core.

Averaging over user-specified pieces is done separately on the real and imaginary parts of the bispectrum, equations (2a) and (2b), and on each autospectrum in the demonimator of equation (3). The averages are then used in calculating bicoherence and biphase.

B. Confidence levels

As each bicoherence is computed, the user may elect to assign it to a bin having a size of his choice. User may also determine the number of bins by specifying the upper bicoherence limit to be considered. After this sorting has been done, program BISCAL determines the bicoherences below which specified fractions ("confidence levels") of the total number of bicoherences calculated bv the program lie. This is done by calculating the fraction of the total which lies below the upper limit of each bin, finding the two closest such fractions which "bracket" the specified confidence level, and linearly interpolating the two corresponding bicoherences to find the one which corresponds to the specified confidence level. A dump of the distribution is available.

USAGE:

(1) Tailoring load module size

For long series of FOURIER coefficients, it is desirable to have large data buffers to keep disc accesses to a minimum. On the other hand these large data buffers can be wasteful for shorter series both in terms of (page · minutes) charges and the penalty in turnaround time. The user is therefore advised to recompile the program to match his needs if he is going to be handling a number of data series of comparable length. To edit the program prior to recompilation, the following steps should be observed:

a. Copy the source file to a duplicate file, and then work with the duplicate file;

b. Change the DIMENSION's of OM1 and OM2 arrays to at least [(L/2 + 2)(NP)], where L = piece length and NP is the number of pieces.

c. Check that array W3 is DIMENSIONed at least 2(NP).

d. Check that SQ is DIMENSIONed at least SQ(L/4 + 1,L/4 + 1) (both dimensions identical).

e. Check that the initialization of variable ISQD is identical to both dimensions of SQ.

f. Recompile.

(2) Control cards

Let aaa = account number

uuu = user number

fff = input file name (RWDISC, keyed)

ttt = file name for transmittal to plotting program (consecutive)

Assume the element file is called BISCALO and exists in account aaa. The Control Card portion of the input deck is then as follows:

```
  !JOB aaa,uuu
* !LIMIT (TIME,4),(CORE,48)
  !ASSIGN F:RAD, (FILE,fff), (DIRECT),(KEYED)
† !ASSIGN F:1, (FILE,ttt),(OUT),(SAVE)
  !LOAD (EF,(BISCALO)), (UNSAT,(3))
  !RUN
  !DATA
  (DATA CARDS)
```

(3) Data cards

Program BISCAL has NAMELIST input followed by one or two data cards depending on the application. The NAMELIST input must be terminated by an * card whether or not any NAMELIST variables are being input.

The NAMELIST parameters and their default values are as follows. In general the switches are to be set to 0 for "off" and 1 for "on".

---

*Time and core limits will vary widely depending on application. See below.

†Necessary only if storing matrix of bicoherences on disc, for transmittal to plotting program BPLOT. DCB assignment may be changed by NAMELIST parameter ISTORE.

NAMELIST VARIABLE	MEANING	DEFAULT VALUE
MF	Number of logical files in input RAD file	21
IDATSTART	Starting location of data in each input logical file	5
INORM	Switch to calculate amplitude-normalized bicoherences	0
LISTBI	Switch to list bicoherences and associated quantities	0
IDUMP	Switch to store bicoherences in a consecutive file for transmittal to plotting program BPLOT	1
ISTORE	DCB assignment for consecutive file of bicoherences created by setting switch IDUMP	1
KONF	Switch for determining bicoherence confidence levels	0
BINSIZE	Bin width for sorting bicoherences. A given bicoherence is placed in bin I, where I = INT (bicoherence/BINSIZE) + 1	0.01
BICOHLIM	The bicoherence "limit" for sorting which determines the number of bins of size BINSIZE, by determining the integer NBINS obtained by rounding (BICOHLIM/ BINSIZE), and then redefining BICOHLIM as (NBINS)(BINSIZE)	1.5
KB1	Switch to display partitioning of bicoherences	1
KB2	Switch to display fraction of total bicoherences under each bin limit	1
IHANN	Determines method of calculating phase error when LISTBI≠0 (See PHSERR, Page 12) IHANN=0 not Hanned and not overlapped IHANN=1 Hanned and overlapped	1

NAMELIST VARIABLE	MEANING	DEFAULT VALUE
SIGLVL	95% confidence level for bicoherences, above which phase error PHSERR is calculated and printed when LISTBI=1	*

The data cards follow the * card which terminates NAMELIST input. They are all generalized, free-field format and are as follows:

Card 1 (Mandatory)

Variable No.	Meaning
1	Logical file for first series†
2	Logical file for second series†
3	Logical file for sum frequency†
4	Total number of pieces in each series
5	Piece length
6, 7	Lower, upper piece sequence numbers inclusive, between which user wants to average
8, 9	Lower, upper limits determining when to print bicoherences when LISTBI≠0. Bicoherences must satisfy: (var. #8) $<$ bicoherence $\leq$ (var. #9)

Card 2 (Input only when KONF≠0)

Variable No.	Meaning
1 (NCONF)	Number of confidence levels to follow ($\leq$10)
2 (CONF(I))	NCONF confidence levels expressed as a decimal, 0 to 1

---

*Default is an approximate value. If P = number of pieces, let the number of degrees of freedom EDOF be defined as follows:

$$EDOF = \begin{cases} 36P^2/(19P-1) & \text{for Hanning, overlapping} \\ 2P & \text{for no Hanning, no overlapping} \end{cases}$$

Then the default value of SIGLVL is

$$SIGLVL = (6/EDOF)^{1/2}$$

†If the first three variables are equal, the program automatically reverts to calculation of auto-bicoherences based on a single file, choosing the minimum necessary domain.

OUTPUT:

(1) Line Printer

Typical line printer output is displayed in Appendix 1. First a dump of NAMELIST parameter values is printed, followed by a facsimile of input data cards. Next, if confidence levels are called for, the "adjusted bicoherence limit" (as determined by the rule given in describing variable BICOHLIM in the section on Data Cards above) is printed.

This is followed by a listing of individual bicoherences if called for. There is one bicoherence value per line, plus other output described below. The listing occurs only within the limits given by variables #8 and #9 on Card 1. The user is cautioned that for cross-bicoherences the total number of bicoherences computed is about 0.75 $(LP)^2$ where LP = piece length. Without sufficiently narrow limits on the bicoherences actually printed, this can result in a very large number of output pages. The items output in each line depend on which form of bicoherence is called for in the program. They are as follows, in order of appearance from left to right on the line:

(A) Default bicoherence (after Yao):

Heading(s)	Meaning
F1 F2 F3	frequency triplet (sequence numbers) associated with this bicoherence
BIC	bicoherence
BIPH	biphase
BIMOD	$\|\langle B(\omega_1,\omega_2)\rangle\|$
BISPECR	Re $\langle B(\omega_1,\omega_2)\rangle$
BISPECI	Im $\langle B(\omega_1,\omega_2)\rangle$
AUTOF1	$\langle\|X(\omega_1)\|^2\rangle$
AUTOF2	$\langle\|X(\omega_2)\|^2\rangle$
AUTOF1F2	$\langle\|X(\omega_3)\|^2\rangle$

Heading(s)	Meaning
SD	$[<\|X(\omega_1)\|^2><\|X(\omega_2)\|^2><\|X(\omega_3)\|^2>]^{1/2}$
PHSERR*	phase error; depends on IHANN (see page 9) $= \sqrt{2P}$ for IHANN=0, P = no. of pieces $= 57.296 \sin^{-1} [1.96/(BIC \cdot W)]$ for IHANN=1 (default), where $W = [36P^2/(19P-1)]^{1/2}$

(B) Alternate bicoherence (normalized by amplitude)

Heading(s)	Meaning
F1 F2 F3	frequency triplet (sequence nubmers) associated with this bicoherence
BIC	bicoherence
BIPH	biphase
BISPPR	Re $<B'(\omega_1,\omega_2)>$
BISPPI	Im $<B'(\omega_1,\omega_2)>$
PHSERR	Defined as above

After listing individual bicoherences, there is an optional listing of how the bicoherences are partitioned among bins, followed by the total number of bicoherences submitted. The format is displayed in Appendix 1.

---

*PHSERR is calculated and printed only above 95% confidence level given by SIGLVL (See NAMELIST variable table). In the line printer output shown in Appendix 1, SIGLVL = 0.5612.

Following this there is an optional display of cumulative fractional numbers of total bicoherence lying below each upper bin limit. Again, the format is displayed in Appendix 1.

Finally, when a calculation of confidence levels is called for, a summary of the result for each confidence level is displayed as in Appendix 1. This is followed by a statement of the number of bicoherences which happened to fall above the limit used in the program (also stated).

(2) Disc

Normally a consecutive disc file is produced by program BISCAL, which is used to transmit the computed bicoherences to program BPLOT for plotting. The first task of program BPLOT is to construct from this file a two-dimensional array whose elements occupy the same relative position in the matrix as do the grid points in the final plot. By using the coding which does this (2 statements for auto, or 5 statements for cross-bicoherences) the user can access this file directly. This will not be delved into here, but the author can be consulted for details.

RESTRICTIONS:

1. The length of the data series which program BISCAL can handle depends on the dimensions of OM1, OM2, and array SQ, as well as on how the data series was processed in the FOURIER transform. The following relations can be used:

Let $\mathcal{L}$ = data series length adjusted so that $2\mathcal{L}$ is evenly divisible by L, the piece length

L = piece length

D = dimensions of OM1 and OM2

NP = number of pieces

50% overlap

$$\mathcal{L} = \frac{L}{2}\left(\frac{2D + L + 4}{L + 4}\right) \text{ and } NP = \frac{2\mathcal{L}}{L} - 1 \qquad (7)$$

$$\mathcal{L} = \frac{2DL}{L + 4} \quad \text{and} \quad NP = \frac{\mathcal{L}}{L} \qquad (8)$$

In both cases the number of points in the series entering BISCAL is (L)(NP), and OM1, OM2 must be dimensioned at least (NP)(L/2 + 2).

The latter requirement has been incorporated in the derivation of expressions (7) and (8).

Example. Program BISCAL has been successfully run with D = 16400 and SQ dimensioned at SQ(70,70). Assuming 50% overlap, eqn. (7) above yields $\mathcal{L}$ = 16256 as the maximum data series length which could have been handled. This is subject to restriction (2) below, which requires that SQ should be dimensioned at least SQ(M,M) where M = L/4 + 1, a condition which is met here.

2. Array W3 must be DIMENSIONed at least 2(NP).
3. The dimension of array SQ must be at least SQ(M,M) where M = L/4 + 1, L = piece length.
4. Variable ISQD must be initialized at either dimension of SQ in (3).
5. L must be an even number with no prime factor greater than 5 and evenly divisible by 4.
6. (No. of confidence levels) $\leq$ 10.
7. (No. of bins used in sorting bicoherences) $\leq$ 200.

STORAGE AND CPU TIME REQUIREMENTS:

As already mentioned economy and convenience may be achieved by tailoring the size of the program to the user's needs if he foresees using it for a number of data series of somewhat comparable length.

The following operating history is provided to assist in this and for general information:

Auto or Cross-Bicoherence	Dimensions of OM1/and OM2	Dimensions of SQ	Piece Length	No. of pieces	Task	Peak core (pg size 512)	cpu time (min)	Charge (CU)
auto	5400	70	256	39	disc file sorting	44	1.895	3.383
auto	5400	70	256	39	disc file	44	1.774	3.034
cross	5400	70	256	39	disc file	44	∿ 6	∿ 5.3

SUBPROGRAMS REQUIRED: None

ERRORS AND DIAGNOSTICS:

Message	Meaning	Action Taken
"(PIECE LENGTH)/2 MUST BE EVEN"	Self-explanatory	abort
"CANNOT HAVE MORE THAN TWO INPUT SERIES"	User has specified 3 distinct logical files for first three entries on Data Card 1	abort

PROGRAMMER: Gerard H. Martineau

ORIGINATOR: Melbourne G. Briscoe

DATE: July 1977

REFERENCES:

(1) Report on Program BPLOT, by G. H. Martineau.

(2) "Handbook for Computer Users," Information Processing Center of W.H.O.I., pp. V-E-1 ff.

(3) Report on Program FOURIER, by G. H. Martineau.

(4) "Bispectral and Cross-Bispectral Analysis of Wind and Currents off Oregon Coast," Ph.D. Thesis at Oregon State University, by N. C. G. Yao, June, 1974.

APPENDIX 1
(BISCAL)

```
 17:43 JAN 26,'78 ID=04F2
JOB 462,1719
LIMIT (TIME,2),(CORE,28),(ORDER),(ACCOUNT)
ASSIGN F:RAD,(FILE,BVEC11),(DIRECT),(KEYED)
LOAD (EF,(BISCAL04R)),(UNSAT,(3))
!P1 ASSOCIATED.


* * ALLOCATION SUMMARY * *

  PROTECTION      LOCATION   PAGES

DATA (00)          A000       7
PROCEDURE (01)     B000       8
DCB (10)           AE00       1
*****************************    SGN= D2L              SIZE=006.4K    ******************************

 ******** PROTECTION TYPES:  00 DATA        01 PROCEDURE   10 STATIC

                            SEGHI=0  AC7B  SEGHI=1  BE61  SEGHI=2  AFFF
                            SEGLO=0  A000  SEGLO=1  B000  SEGLO=2  AE00

                            00 SIZE=  C7C  01 SIZE=  E62  10 SIZE=  200
RUN
 IDATSTART = 5
 MF = 21
 ISTORE = 1
 BINSIZE = 1.000000E-02
 BICOHLIM = 1.50000
 IHI = 1
 KB1 = 1
 KB2 = 1
 KONF = 1
 LISTBI = 1
 IDUMP = 0
 INORM = 0
 IHANN = 1
 1  1  1  10  64  1  10  .545  .575
 4  .90  .95  .99  .999
ADJUSTED BICOHERENCE LIMIT, HOLDING BINSIZE AS INPUT   1.50

  F1    F2    F3    BIC    BIPH     BIMOD      BISPECR     BISPECI    AUTOF1     AUTOF2     AUTOF1F2     SD      PHSERR

   6     2     8   .568     73   .610E-02   .183E-02   .582E-02   .558E-01   .353E-01   .586E-01   .107E-01   52.2
   6     5    11   .546    114   .745E-02  -.301E-02   .681E-02   .558E-01   .561E-01   .594E-01   .136E-01
   8     4    12   .553     35   .623E-02   .508E-02   .361E-02   .586E-01   .432E-01   .502E-01   .113E-01
  20     9    29   .567   -150   .135E-01  -.117E-01  -.667E-02   .113E 00   .874E-01   .574E-01   .238E-01   52.3

                                BIN NO      BIN LIMITS       NO OF BIC'S

                                  1       .00      .01            1
                                  2       .01      .02            0
                                  3       .02      .03            1
                                  4       .03      .04            4
                                  5       .04      .05            2
                                  6       .05      .06            4
                                  7       .06      .07            3
                                  8       .07      .08            0
                                  9       .08      .09            4
```

BIN NO	BIN LIMITS		NO OF BIC'S
10	.09	.10	2
11	.10	.11	3
12	.11	.12	2
13	.12	.13	9
14	.13	.14	4
15	.14	.15	4
16	.15	.16	4
17	.16	.17	6
18	.17	.18	6
19	.18	.19	11
20	.19	.20	8
21	.20	.21	4
22	.21	.22	5
23	.22	.23	8
24	.23	.24	1
25	.24	.25	5
26	.25	.26	4
27	.26	.27	7
28	.27	.28	12
29	.28	.29	8
30	.29	.30	6
31	.30	.31	7
32	.31	.32	2
33	.32	.33	9
34	.33	.34	5
35	.34	.35	5
36	.35	.36	8
37	.36	.37	8
38	.37	.38	2
39	.38	.39	3
40	.39	.40	9
41	.40	.41	4
42	.41	.42	5
43	.42	.43	2
44	.43	.44	7
45	.44	.45	5
46	.45	.46	4
47	.46	.47	3
48	.47	.48	1
49	.48	.49	1
50	.49	.50	3
51	.50	.51	2
52	.51	.52	1
53	.52	.53	1
54	.53	.54	1
55	.54	.55	2
56	.55	.56	1
57	.56	.57	2
58	.57	.58	1
59	.58	.59	1
60	.59	.60	3
61	.60	.61	1
62	.61	.62	0
63	.62	.63	0
64	.63	.64	1
65	.64	.65	1
66	.65	.66	0
67	.66	.67	0
68	.67	.68	0
69	.68	.69	0

70	.69	.70	0
71	.70	.71	1
72	.71	.72	1
73	.72	.73	1
74	.73	.74	0
75	.74	.75	0
76	.75	.76	0
77	.76	.77	0
78	.77	.78	0
79	.78	.79	0
80	.79	.80	0
81	.80	.81	0
82	.81	.82	0
83	.82	.83	1
84	.83	.84	1
85	.84	.85	0
86	.85	.86	0
87	.86	.87	0
88	.87	.88	0
89	.88	.89	1
90	.89	.90	0
91	.90	.91	0
92	.91	.92	0
93	.92	.93	0
94	.93	.94	0
95	.94	.95	0
96	.95	.96	1
97	.96	.97	0
98	.97	.98	0
99	.98	.99	0
100	.99	1.00	0
101	1.00	1.01	0
102	1.01	1.02	0
103	1.02	1.03	0
104	1.03	1.04	0
105	1.04	1.05	0
106	1.05	1.06	0
107	1.06	1.07	0
108	1.07	1.08	0
109	1.08	1.09	0
110	1.09	1.10	0
111	1.10	1.11	0
112	1.11	1.12	0
113	1.12	1.13	0
114	1.13	1.14	0
115	1.14	1.15	0
116	1.15	1.16	0
117	1.16	1.17	0
118	1.17	1.18	0
119	1.18	1.19	0
120	1.19	1.20	0
121	1.20	1.21	0
122	1.21	1.22	0
123	1.22	1.23	0
124	1.23	1.24	0
125	1.24	1.25	0
126	1.25	1.26	0
127	1.26	1.27	0
128	1.27	1.28	0
129	1.28	1.29	0

BIN NO	BIN LIMITS		NO OF BIC'S
130	1.29	1.30	0
131	1.30	1.31	0
132	1.31	1.32	0
133	1.32	1.33	0
134	1.33	1.34	0
135	1.34	1.35	0
136	1.35	1.36	0
137	1.36	1.37	0
138	1.37	1.38	0
139	1.38	1.39	0
140	1.39	1.40	0
141	1.40	1.41	0
142	1.41	1.42	0
143	1.42	1.43	0
144	1.43	1.44	0
145	1.44	1.45	0
146	1.45	1.46	0
147	1.46	1.47	0
148	1.47	1.48	0
149	1.48	1.49	0
150	1.49	1.50	0

TOTAL NO OF BIC'S SUBMITTED: 256

BIN NO	BIN LIMITS		FRACT BELOW UPPER BIN LIM
1	.00	.01	.004
2	.01	.02	.004
3	.02	.03	.008
4	.03	.04	.023
5	.04	.05	.031
6	.05	.06	.047
7	.06	.07	.059
8	.07	.08	.059
9	.08	.09	.074
10	.09	.10	.082
11	.10	.11	.094
12	.11	.12	.102
13	.12	.13	.137
14	.13	.14	.152
15	.14	.15	.168
16	.15	.16	.184
17	.16	.17	.207
18	.17	.18	.230
19	.18	.19	.273
20	.19	.20	.305
21	.20	.21	.320
22	.21	.22	.340
23	.22	.23	.371
24	.23	.24	.375
25	.24	.25	.395
26	.25	.26	.410
27	.26	.27	.438
28	.27	.28	.484
29	.28	.29	.516
30	.29	.30	.539
31	.30	.31	.566
32	.31	.32	.574

33	.32	.33	.609
34	.33	.34	.629
35	.34	.35	.648
36	.35	.36	.680
37	.36	.37	.711
38	.37	.38	.719
39	.38	.39	.730
40	.39	.40	.766
41	.40	.41	.781
42	.41	.42	.801
43	.42	.43	.809
44	.43	.44	.836
45	.44	.45	.855
46	.45	.46	.871
47	.46	.47	.883
48	.47	.48	.887
49	.48	.49	.891
50	.49	.50	.902
51	.50	.51	.910
52	.51	.52	.914
53	.52	.53	.918
54	.53	.54	.922
55	.54	.55	.930
56	.55	.56	.934
57	.56	.57	.941
58	.57	.58	.945
59	.58	.59	.949
60	.59	.60	.961
61	.60	.61	.965
62	.61	.62	.965
63	.62	.63	.965
64	.63	.64	.969
65	.64	.65	.973
66	.65	.66	.973
67	.66	.67	.973
68	.67	.68	.973
69	.68	.69	.973
70	.69	.70	.973
71	.70	.71	.977
72	.71	.72	.980
73	.72	.73	.984
74	.73	.74	.984
75	.74	.75	.984
76	.75	.76	.984
77	.76	.77	.984
78	.77	.78	.984
79	.78	.79	.984
80	.79	.80	.984
81	.80	.81	.984
82	.81	.82	.984
83	.82	.83	.988
84	.83	.84	.992
85	.84	.85	.992
86	.85	.86	.992
87	.86	.87	.992
88	.87	.88	.992
89	.88	.89	.996
90	.89	.90	.996
91	.90	.91	.996
92	.91	.92	.996

BIN NØ	BIN LIMITS		FRACT BELØW UPPER BIN LIM
93	.92	.93	.996
94	.93	.94	.996
95	.94	.95	.996
96	.95	.96	1.000
97	.96	.97	1.000
98	.97	.98	1.000
99	.98	.99	1.000
100	.99	1.00	1.000
101	1.00	1.01	1.000
102	1.01	1.02	1.000
103	1.02	1.03	1.000
104	1.03	1.04	1.000
105	1.04	1.05	1.000
106	1.05	1.06	1.000
107	1.06	1.07	1.000
108	1.07	1.08	1.000
109	1.08	1.09	1.000
110	1.09	1.10	1.000
111	1.10	1.11	1.000
112	1.11	1.12	1.000
113	1.12	1.13	1.000
114	1.13	1.14	1.000
115	1.14	1.15	1.000
116	1.15	1.16	1.000
117	1.16	1.17	1.000
118	1.17	1.18	1.000
119	1.18	1.19	1.000
120	1.19	1.20	1.000
121	1.20	1.21	1.000
122	1.21	1.22	1.000
123	1.22	1.23	1.000
124	1.23	1.24	1.000
125	1.24	1.25	1.000
126	1.25	1.26	1.000
127	1.26	1.27	1.000
128	1.27	1.28	1.000
129	1.28	1.29	1.000
130	1.29	1.30	1.000
131	1.30	1.31	1.000
132	1.31	1.32	1.000
133	1.32	1.33	1.000
134	1.33	1.34	1.000
135	1.34	1.35	1.000
136	1.35	1.36	1.000
137	1.36	1.37	1.000
138	1.37	1.38	1.000
139	1.38	1.39	1.000
140	1.39	1.40	1.000
141	1.40	1.41	1.000
142	1.41	1.42	1.000
143	1.42	1.43	1.000
144	1.43	1.44	1.000
145	1.44	1.45	1.000
146	1.45	1.46	1.000
147	1.46	1.47	1.000
148	1.47	1.48	1.000
149	1.48	1.49	1.000
150	1.49	1.50	1.000

CØNFIDENCE

LEVEL (PERCENT)	BICOHERENCE
90.0	.498
95.0	.591
99.0	.834
99.9	.957

NO. OF BICOHERENCES GREATER THAN SPECIFIED MAXIMUM OF 1.50 IS 0

STOP NORMAL PROGRAM COMPLETION

NAME: BPLOT

PURPOSE: To construct auto or cross-bicoherence plot from disc file produced by program BISCAL[1]

MACHINE: Xerox Sigma-7

SOURCE LANGUAGE: FORTRAN IV

DESCRIPTION:

Program BPLOT reads a consecutive file of auto or cross-bicoherences produced by program BISCAL and then makes a plot of these bicoherences over the appropriate domain.

USAGE:

(1) Control cards

Let aaa = account number
uuu = user number
fff = input file name
ppp = plot file name

Assume element file BPLOTR (object of BPLOT) is in account aaa. Assume any plotting is on Versatec or graphics terminal. Then the Control Card portion of the input deck is:

```
 !JOB aaa,uuu
 !LIMIT (TIME,2),(CORE,38)
 !ASSIGN F:1,(FILE,fff)
*!ASSIGN F:95,(FILE,ppp),(OUT),(SAVE)
 !LOAD (EF,(BPLOTR),(PLOTDFER,3)),(UNSAT,(3))
 !RUN
 !DATA
──────────→ (DATA CARDS)
†!PLOTV
```

* - Necessary only if plot file is to be saved after this job.

† - Necessary only for a Versatec plot.

(2) Data Cards

Program BPLOT has NAMELIST input followed by three data cards. The NAMELIST input must be terminated by an * card whether or not any NAMELIST variables are being input.

The NAMELIST parameters and their default values are as follows:

NAMELIST variable	Meaning	Default Value
ISTORE	DCB assignment for input consecutive file	1
NDECPL	Format control for axis annotation: >0: Number of digits to the right of the decimal point which are plotted, after proper rounding =0: Only integer portion of number and decimal point are plotted, after rounding =-1: Only integer portion of number is plotted, after rounding <-1: |NDECPL|-1 digits are truncated from the integer portion after rounding	1
WIDTH	Width, in inches, of plot of L/2 points, where L=pc. length	8.75
CMPPT	Spacing in centimeters between adjacent data (grid) points on plot. If entered by user, this will override any specification of variable WIDTH	WIDTH*2.54/(L/2) (L=pc. length)
IHISTORY	Switch to include information about data origin on plot; i.e., original buoy format file names and variable numbers, and to print processing history on plot (from program FOURIER) IHISTORY=1 for "yes", 0 for "no"	1

The data cards follow the * card which terminates NAMELIST input. They are all in generalized, free-field format and are as follows:

Card 1

Variable No.	Meaning
1	1 for auto-bicoherence 2 for cross-bicoherence
2	piece length

Card 2

Variable No.	Meaning
1 (SAMPSEC)	Sample interval in seconds
2 (FREQTIC)	*Distance between tic marks on plot in $(hr)^{-1}$
3 (NCONLV)	Number of contour levels to follow (≤ 10)
4 thru (3+NCONLV)	NCONLV contour levels for plot
(4+NCONLV) thru (3+2·NCONLV)	NCONLV confidence levels (percent) for contour levels

Card 3

Up to 72 characters of identification, which will appear immediately above the plot. The first 36 characters will appear on one line, and the next 36 characters on the next line. The user is responsible for a proper transition of text from one line to the next, occurring between columns 36 and 37.

OUTPUT:

(1) Line Printer

Sample line printer output appears in Appendix 1. A facsimile of the three input cards is printed, followed by a listing of the contour levels from Card 2 under a separate heading, "CONTOUR LEVELS."

(2) Plots

Either a disc plot file or a Calcomp plot tape can be output by the program. See Ref. 2 for the latter. By using the PLOTV or PLOTC processor, Versatec or Calcomp plots can be produced either in the same job as BPLOT or separately. Additionally, the Versatec plot file can be displayed on a graphics terminal via the PLOTT processor.

Two sample reduced Calcomp plots constitute Appendices 2 and 3, which display auto and cross-bicoherences respectively. The notation is self-explanatory.

*Full-scale frequency (Nyquist frequency) is 1800/SAMPSEC. Variable FREQTIC does not have to divide evenly into the full-scale frequency.

RESTRICTIONS:

(1) Program BPLOT uses a large two-dimensional buffer, "DATA", which can be thought of as a matrix whose elements correspond one-for-one with the grid points of the plot, and whose rectangular "boundary" completely contains the plot, the non-data points being padded with flags to that effect. The dimensions of DATA therefore govern the maximum piece length which can be plotted. The rule is as follows: If L is the piece length, then array DATA must be dimensioned at least DATA(R,C) where:

$$C \geq L/2$$

$$R \geq 1.5C + 1$$

The "standard" dimensions of DATA are DATA(200,130). If recompilation is necessary, then the only other change which must be made is to the initialization of variables MD1 and MD2 in the statement following the DIMENSION statement. MD1, MD2 must be the first and second dimensions of DATA respectively.

(2) The number of contour levels must be ≤ 15.

SUBPROGRAMS REQUIRED: PLOTDFER, for a Versatec or graphics terminal plot file.

STORAGE REQUIREMENTS: In a typical BPLOT run in which BPLOT plus PLOTDFER were loaded, and array DATA in BPLOT was dimensioned DATA (200,130), a peak core of 75 512-word pages were used.

TIMING: In the run just cited, 0.397 min of CPU time was used.

ERRORS AND DIAGNOSTICS: None

PROGRAMMER: GERARD H. MARTINEAU

ORIGINATOR: MELBOURNE G. BRISCOE

DATE: July 1977

REFERENCES: (1) Report on Program BISCAL, by Gerard H. Martineau, Woods Hole Oceanographic Inst.
(2) Handbook for Computer Users, Information Processing Center of Woods Hole Oceanographic Inst.

APPENDIX 1
(BPLOT)

```
 13:46 DEC 30,'77 ID=0929
JOB 460,1719
LIMIT (TIME,2),(CORE,38)
ASSIGN F:1,(FILE,TIGOUT)
ASSIGN F:95,(FILE,BISC1),(OUT),(SAVE)
LOAD (EF,(BPLOTR),(PLOTDFER,3)),(UNSAT,(3))
:P1 ASSOCIATED.


* * ALLOCATION SUMMARY * *

  PROTECTION       LOCATION   PAGES

DATA (00)           A000       3C
PROCEDURE (01)     11A00        6
DCB (10)           11800        1
******************************     SGN= IZL                SIZE=032.3K     ******

 ******** PROTECTION TYPES:  00 DATA          01 PROCEDURE    10 STATIC

                             SEGHI=0 116F9   SEGHI=1 1254B   SEGHI=2 119FF
                             SEGLO=0  A000   SEGLO=1 11A00   SEGLO=2 11800

                             00 SIZE= 76FA   01 SIZE=  B4C   10 SIZE=  200
RUN
 1 256
 225 0.8 3 .42 .52 .637 95. 99. 99.9
 IWEX B10EAST

CONTOUR LEVELS:
      .420           .520           .637
 *STOP* NORMAL PROGRAM COMPLETION
```

APPENDIX 2
(BPLOT)

AUTO BICOHERENCE

ORIGINAL FILE NAMES:		VARIABLE NUMBERS:
FREQUENCY 1:	GENRANOUTPUT	0
FREQUENCY 2:	GENRANOUTPUT	0
FREQUENCY 3:	GENRANOUTPUT	0

PROCESSING HISTORY

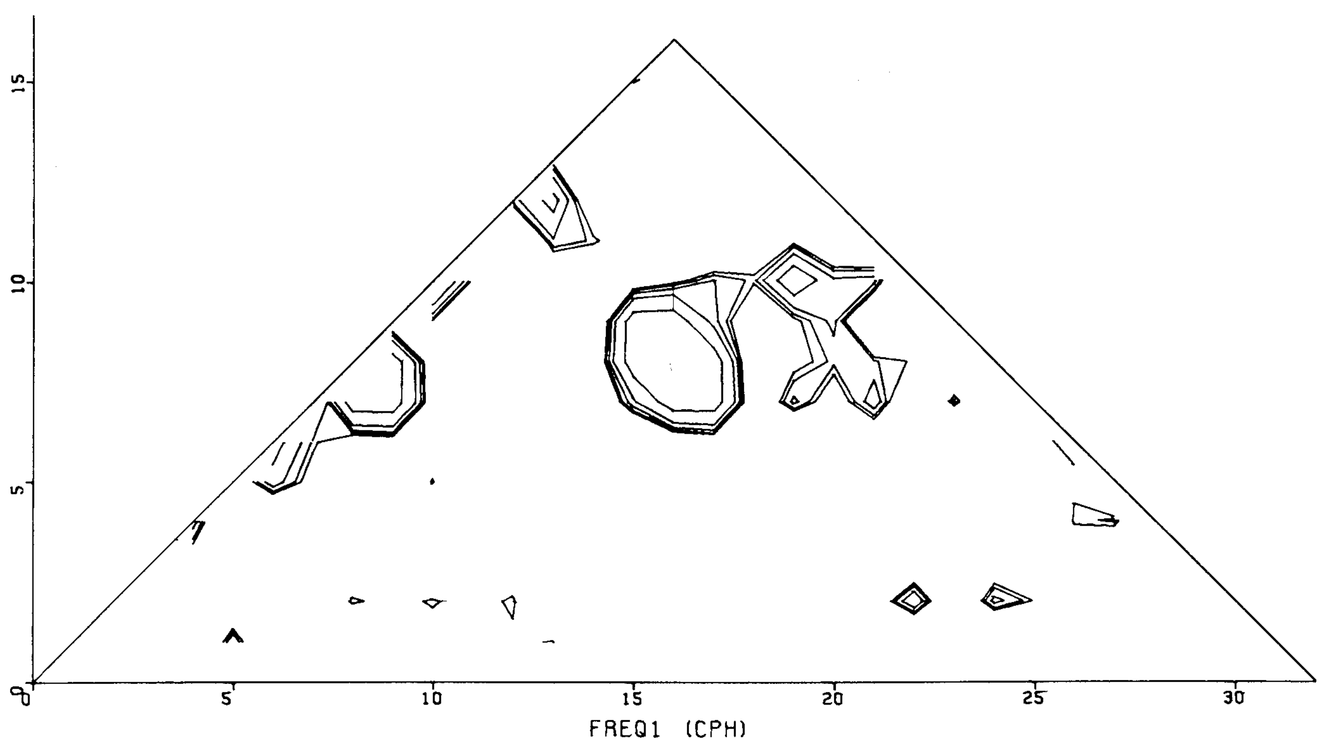
NO LF'S 8*NO WDS DATA 352*PREW 0*SUBSAMP 0*NSUBSAMP 0*PC SIZE 64
*NO PCS 10*OLAP 1*ISUBT 0*HANN 1 *CREATED 19:20 DEC 21, '77

TIME OF PLOT:
09:06 DEC 22, '77

CONTOUR LEVELS AND PERCENT CONFIDENCE

0.420	80.0
0.447	85.0
0.498	90.0
0.591	95.0
0.834	99.0

S8+S16+S24 NYQUIST=32 HANNED
FILE BVPL11 S/N=10



APPENDIX 3
(BPLOT)

CROSS BICOHERENCE

ORIGINAL FILE NAMES: VARIABLE NUMBERS:

FREQUENCY 1:	GENRANOUTPUT	0
FREQUENCY 2:	GENRANOUTPUT	0
FREQUENCY 3:	GENRANOUTPUT	0

PROCESSING HISTORY

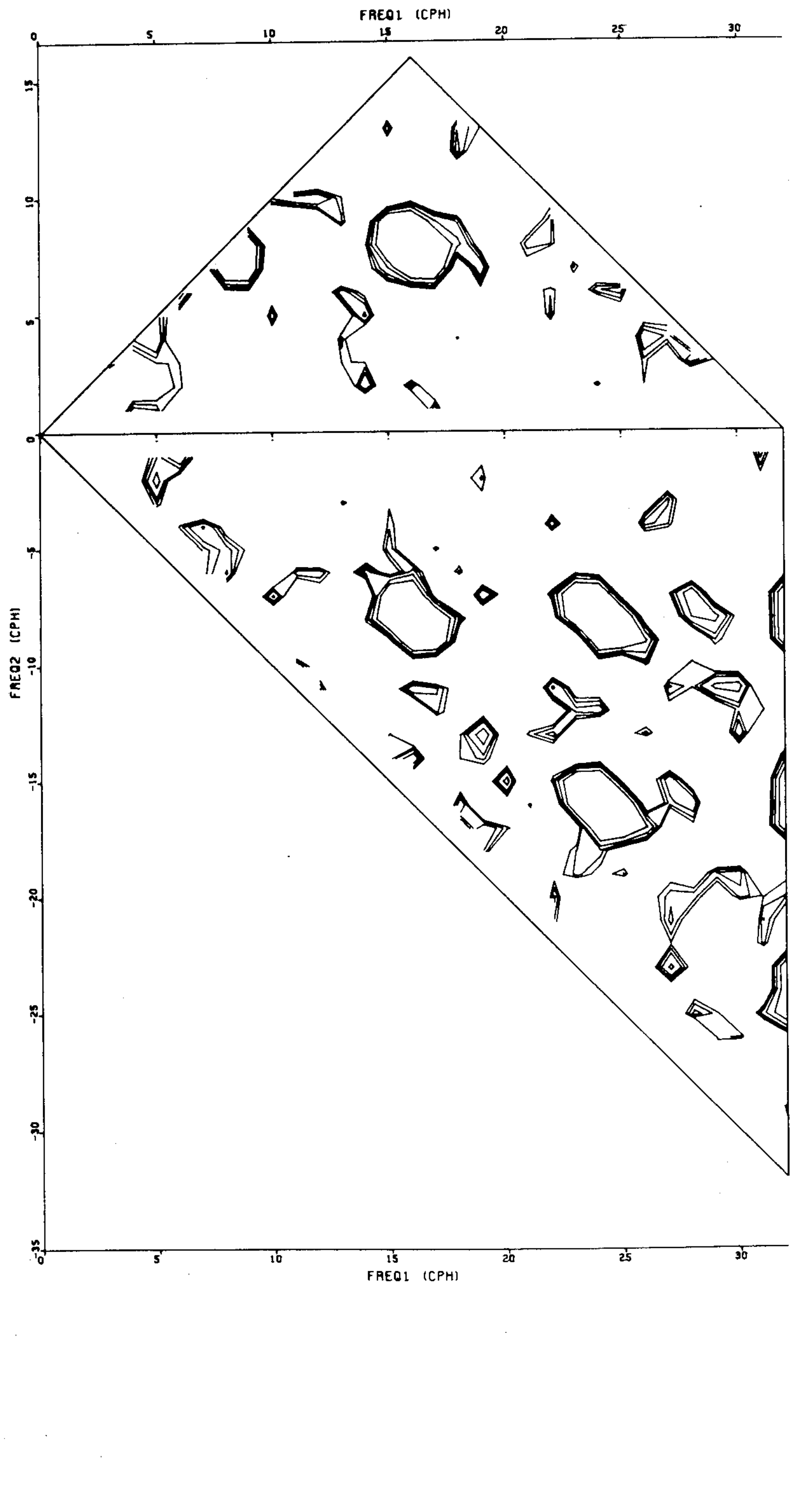
NO LF'S 4=NO HOS DATA 352=PREW 0=SUBSAMP 0=NSUBSAMP 0=PC SIZE 64
=NO PCS 10=OLAP 1=ISUBT 0=HANN 1 =CREATED 17:18 DEC 15,'77

CONTOUR LEVELS AND PERCENT CONFIDENCE

TIME OF PLOT:
09:38 DEC 22,'77

0.400	80.0
0.431	85.0
0.468	90.0
0.551	95.0
0.675	99.0

X=C8+C16+C24;Y=-S8-S16-S24;FORM XXY
NYQ=32 HANNED S/N=10 FILE BVPL3



NAME: BISUM

PURPOSE: To integrate certain bispectral quantities along paths of constant sum frequency in the $\omega_1 - \omega_2$ plane.

MACHINE: Xerox Sigma-7

SOURCE LANGUAGE: FORTRAN IV

DESCRIPTION:

Program BISUM is similar to the bispectral program BISCAL,[1] and the user is referred to the report on BISCAL for computational procedures. The difference in the two programs is that the sorting capability of BISCAL has been replaced in BISUM by summations of bicoherences and other quantities related to bispectra, over paths of constant sum frequencies (ω_3) in the $\omega_1 - \omega_2$ plane. The paths are diagonal straight lines of slope (-1). The summing is done separately above and below the ω_1 axis; i.e., for positive and negative ω_2. A formatted listing of these integrals is presented. The integrals may additionally be written to a disc file. It is also possible to write the same file of bicoherences that is written by BISCAL for transmittal to the plotting program BPLOT.[2]

INPUT:

Input consists of the same type of RWDISC file of Fourier coefficients as is used by program BISCAL.[1]

OUTPUT: (I) Line Printer

A sample of the printed output appears in Appendix 1. First the values of all NAMELIST variables are listed, along with a facsimile of the input data card. This is followed by a formatted list of integrals.

The integrals presented in the case of calculation of bicoherences by Form 1 as described in the program BISCAL report, are as follows:

Let N be the number of points over the path of integration.

(1) BR : real part of bispectrum at frequency ω_3

(2) BR/N

(3) BI : imaginary part of bispectrum at frequency ω_3

(4) BI/N

(5) BIC : bicoherence[1]

(6) BIC/N

(7) $(\text{BIC})^2$

(8) $(\text{BIC})^2/\text{N}$

(9) $S(\omega_3)*(\text{BIC})^2$ where $S(\omega_3)$ is square of auto-spectrum at frequency ω_3

(10) $(S(\omega_3)*(\text{BIC})^2)/\text{N}$

In the case of calculation of bicoherences of Form 2 (from amplitude-normalized bispectrum) in the program BISCAL report, all of the above quantities corresponding to the amplitude-normalized bispectrum are calculated, except (9) and (10) which are listed as zero.

It will be noted that both the sum frequency sequence number ($|\omega_3|$) and the number of independent variable values that enter each integral are printed for each line of output, each line corresponding to a single value of $|\omega_3|$.

If L is the piece length, there are (L/2-1) lines of output for an auto-bispectrum and twice that many for a cross-bispectrum.

The user may also get a listing of bicoherences and related quantities between limits which he can set. This listing is described in the report on program BISCAL.[1] The limits are set in a data card to be described in the "USAGE" section.

(II) Disc

Two types of disc files may be written by program BISUM. The first is the same type of file of bicoherences written by BISCAL[1] for transmittal to plotting program BPLOT. The second is a consecutive file containing all integrals computed by BISUM. This file is organized as follows (L = piece length):

cross-bispectrum {
  auto-bispectrum {
    1 EBCID header record followed by
    (L/2-1) binary records followed by
  }
  (L/2-1) binary records
}

This file should be read by a FORTRAN formatted READ statement (for the header record, to fill an integer array dimensioned 33), followed by one FORTRAN binary READ statement for each line of printed output. Each READ statement should read 13 words. If the user wishes to access words 1, 2, or 13, he is notified that these are integer type, all others being real.

USAGE: (I) Input Card Deck

1) Control Cards

Let aaa = account number
uuu = user number
INFILE = input file of FOURIER coefficients
OUTBIC = output file of bicoherences
OUTSUMS = output file of integrals
BISUMR = object file of program BISUM in account aaa

The control card deck can appear as follows:

```
     !JOB aaa,uuu
  *  !LIMIT (TIME,4),(CORE,50)
     !ASSIGN F:RAD,(FILE,INFILE),(DIRECT),(KEYED)
  †  !ASSIGN F:1,(FILE,OUTBIC),(OUT),(SAVE)
 ††  !ASSIGN F:2,(FILE,OUTSUMS),(OUT),(SAVE)
     !LOAD (EF,(BISUMR)),(UNSAT,(3))
     !RUN
     !DATA
     ----------> Data cards
```

---

*Core limit can vary greatly with dimensioning of working arrays. See below.

†Necessary only when outputting file of bicoherences.

††Necessary only when outputting file of integrals.

2) Data Cards

Data card input format is similar to that for program BISCAL.[1] There is NAMELIST input, followed by an * card whether or not NAMELIST parameters are entered, followed by a single data card in free-field, generalized format.

The NAMELIST parameters follow, with their default values. "Switches" are set to 0 for "off", 1 for "on":

Variable	Meaning	Default Value
MF	Number of logical files in input RWDISC file of FOURIER coefficients	21
IDATSTART	Location of first word of data in each logical file of input file	5
INORM	Switch to calculate amplitude-normalized bicoherences	0
LISTBI	Switch to list bicoherences and related quantities[1] within user-specified limits	0
IDUMP	Switch to output file of bicoherences for transmittal to plotting program BPLOT	0
ISTORE	DCB assignment for output file of bicoherences created by setting switch IDUMP	1
KDISC	Switch to output file of integrals	0
KDCB	DCB assignment for output file of integrals	2
IHANN	Determines method of calculating phase error when LISTBI≠0 (See PHSERR on page 12 of BISCAL report). IHANN = 0 not Hanned and not overlapped IHANN=1 Hanned and overlapped	1
SIGLVL	95% confidence level for bicoherences, above which phase error PHSERR is calculated and printed when LISTBI=1	†

The single data card, which follows the * NAMELIST terminator card has the following entries in free-field, generalized format:

---

†Default is an approximate value. If P = number of pieces, let the number of degrees of freedom EDOF be defined as follows:

$$EDOF = \begin{cases} 36P^2/(19P-1) & \text{for Hanning, overlapping} \\ 2P & \text{for no Hanning, no overlapping} \end{cases}$$

Then the default value of SIGLVL is:

$$SIGLVL = (6/EDOF)^{1/2}$$

Variable No.	Meaning
*1	Logical file in input RWDISC file associated with frequency ω_1
*2	Logical file associated with ω_2
*3	Logical file associated with $\omega_3 = -\omega_1 - \omega_2$
4	Number of pieces in input series of Fourier coefficients
5	Piece length
6, 7	Two piece sequence numbers between which it is desired to average, inclusively
8, 9	Lower, upper limits determining when to print bicoherence when LISTBI$\neq$0. Bicoherences must satisfy: (var. #9) < bicoherence $\leq$ (var. #10)

(II) Altering Load Module Size

As in BISCAL, program BISUM attempts to minimize disc accesses in the same way.[1] This involves the use of two large buffers which largely determine the size of the load module. There may be worthwhile benefits to the user, in terms of page-minute charges and turn-around time, in recompiling the program using buffers which are no longer than needed.

The user is therefore advised to recompile the program to match his needs if he is going to be handling a number of data series of comparable length. To edit the program prior to recompilation, the following steps should be observed:

a. Copy the source file to a duplicate file, and then work with the duplicate file;

b. Change the DIMENSION's of OM1 and OM2 arrays to at least [(L/2+2)(NP)], where L = piece length and NP is the number of pieces.

---

*If variables 1, 2, 3 are all equal, then auto-bispectra will be computed and the minimum necessary doman will be used.

c. Check that array W3 is DIMENSIONed at least 2(NP).

d. Check that SQ is DIMENSIONed at least SQ(L/4 + 1, L/4 + 1) (both dimensions identical).

e. Check that the initialization of variable ISQD is identical to both dimensions of SQ.

f. There are ten arrays used to accumulate sums of bicoherences and related quantities. These must be all DIMENSIONed at least (L/2). The arrays are BRU, BIU, BICU, BIC2U, BCSU, BRL, BIL, BICL, BIC2L, and BCSL.

g. Variable JDIM must be initialized at the dimension of the arrays in (f).

h. Recompile.

A further reduction in size of the program can be achieved if the user does not wish to produce a file of bicoherences. Array SQ has two equal dimensions, which must be at least (L/2+1) if this file is produced. If there is to be no file of bicoherences, then SQ can be dimensioned SQ(2,2). Variable ISQD must be initialized at one of the dimensions of SQ.

RESTRICTIONS:

1. The length of the data series which program BISUM can handle depends on the dimensions of OM1, OM2, and array SQ, as well as on how the data series was processed in the FOURIER transform. The following relations can be used:

Let $\mathcal{L}$ = data series length adjusted so that $2\mathcal{L}$ is evenly divisible by L, the piece length

L = piece length

D = dimensions of OM1 and OM2

NP = number of pieces

50% overlap

$$\mathcal{L} = \frac{L}{2}\left(\frac{2D + L + 4}{L + 4}\right) \text{ and } NP = \frac{2\mathcal{L}}{L} - 1 \qquad (1.)$$

$$\mathcal{L} = \frac{2DL}{L + 4} \text{ and } NP = \frac{\mathcal{L}}{L} \qquad (2.)$$

In both cases the number of points in the series entering BISUM is (L)(NP), and OM1, OM2 must be dimensioned at least (NP)(L/2 + 2).

The latter requirement has been incorporated in the derivation of expressions 1. and 2.

Example. Program BISUM has been successfully run with D = 16400 and SQ dimensioned at SQ(70,70). Assuming 50% overlap, Eq. (7) above yields $\mathcal{L}$ = 16256 as the maximum data series length which could have been handled. This is subject to restriction (2) below, which requires that SQ should be dimensioned at least SQ(M,M) where M = L/4 + 1, a condition which is met here.

2. Array W3 must be DIMENSIONed at least 2(NP).

3. The dimension of array SQ must be at least SQ(M,M) where M = L/4 + 1, L = piece length.

4. Variable ISQD must be initialized at either dimension of SQ in (3).

5. The arrays used to accumulate sums of bicoherences and related quantities must all be DIMENSIONed at least (L/2). These arrays are BRU, BIU, BICU, BIC2U, BCSU, BRL, BIL, BICL, BIC2L, and BCSL.

6. Variable JDIM must be initialized to the dimension of the arrays in 5.

7. L must be an even number with no prime factor greater than 5, and evenly divisible by 4.

SUBPROGRAMS REQUIRES: None

ERRORS AND DIAGNOSTICS:

Message	Meaning	Action Taken
"(PIECE LENGTH)/2 MUST BE EVEN"	Piece length must be evenly divisible by four	Abort
"CANNOT HAVE MORE THAN TWO INPUT SERIES"	Three distinct logical files have been referenced by first three entries on data card	Abort

PROGRAMMER: GERARD H. MARTINEAU

ORIGINATOR: MELBOURNE G. BRISCOE

DATE: July, 1977

REFERENCES:

1. Report on Progrqm BISCAL, by G. H. Martineau, Woods Hole Oceanographic Institution.

2. Report on Program BPLOT, by G. H. Martineau, Woods Hole Oceanographic Insitution.

APPENDIX 1
(BISUM)

```
 15:18 JAN 26,'78 ID=0462
JOB 462,1719
LIMIT (TIME,2),(CORE,35),(ORDER),(ACCOUNT)
ASSIGN F:RAD,(FILE,BVEC11),(DIRECT),(KEYED)
LOAD (EF,(BISUM02R)),(UNSAT,(3))
:P1 ASSOCIATED.


* * ALLOCATION SUMMARY * *

  PROTECTION      LOCATION   PAGES

DATA (00)          A000       6
PROCEDURE (01)     B000       8
DCB (10)           AC00       2
******************************   SGN= DSL                  SIZE=007.0K   *****

 ******** PROTECTION TYPES:  00 DATA         01 PROCEDURE    10 STATIC

                             SEGHI=0  ABBD   SEGHI=1  BF65   SEGHI=2  AFFF
                             SEGLO=0  A000   SEGLO=1  B000   SEGLO=2  AC00

                             00 SIZE=  BBE   01 SIZE=  F66   10 SIZE=  400
RUN
 IDATSTART = 5
 MF = 21
 ISTORE = 0
 BINSIZE = 1.000000E-02
 BICOHLIM = 1.50000
 IHI = 1
 KB1 = 1
 KB2 = 1
 LISTBI = 0
 IDUMP = 0
 INORM = 0
 IHANN = 1
 KDISC = 0
 KDCB = 2
 4  4  7  10  64  1  10  .0001  1000.
```

INTEGRALS OVER POSITIVE FREQUENCIES ALONE

SUM FREQ SEQ NO	NO IN SCAN (N)	BR	BR/N	BI	BI/N	BIC	BIC/N	BIC**2	(BIC**2)/N	S(W3)* BIC**2	(S(W3)* BIC**2)/N	SUM FREQ SEQ NO
2	1	-.720E-02	-.720E-02	.191E-01	.191E-01	.651	.651	.424	.424	.149E-01	.149E-01	2
3	1	.181E-02	.181E-02	.247E-02	.247E-02	.156	.156	.243E-01	.243E-01	.926E-03	.926E-03	3
4	2	-.452E-03	-.226E-03	-.857E-02	-.428E-02	.520	.260	.152	.761E-01	.657E-02	.329E-02	4
5	2	-.167E-01	-.837E-02	-.388E-02	-.194E-02	.750	.375	.315	.158	.177E-01	.885E-02	5
6	3	-.362E-02	-.121E-02	.144E-01	.479E-02	1.15	.383	.467	.156	.261E-01	.869E-02	6
7	3	.149	.497E-01	.491E-01	.164E-01	.750	.250	.197	.658E-01	3.28	1.09	7
8	4	-.816	-.204	1.59	.398	1.45	.363	.573	.143	38.5	9.61	8
9	4	-1.09	-.272	2.49	.622	1.29	.323	.508	.127	8.54	2.14	9
10	5	-.758E-02	-.152E-02	-.195	-.391E-01	1.26	.251	.424	.849E-01	.250E-01	.499E-02	10
11	5	-.208	-.416E-01	.239	.479E-01	1.27	.254	.357	.713E-01	.212E-01	.424E-02	11
12	6	-.287	-.478E-01	.234E-01	.391E-02	1.85	.309	.742	.124	.372E-01	.621E-02	12
13	6	-.186	-.310E-01	-.653E-01	-.109E-01	1.10	.183	.245	.408E-01	.147E-01	.246E-02	13
14	7	-.194	-.277E-01	.465	.665E-01	1.47	.210	.413	.589E-01	.223E-01	.319E-02	14
15	7	-1.14	-.163	135.	19.3	2.43	.347	1.43	.204	24.3	3.47	15
16	8	-16.2	-2.02	684.	85.5	3.69	.461	2.62	.328	176.	22.0	16
17	8	-4.74	-.592	138.	17.3	2.55	.319	1.48	.185	24.5	3.06	17
18	9	.192	.213E-01	-1.22	-.135	2.48	.276	.821	.912E-01	.344E-01	.383E-02	18
19	9	-.231	-.257E-01	.854E-01	.949E-02	2.27	.252	.838	.931E-01	.485E-01	.539E-02	19
20	10	-.279	-.279E-01	-.191E-01	-.191E-02	2.13	.213	.613	.613E-01	.691E-01	.691E-02	20
21	10	-.214	-.214E-01	-.110	-.110E-01	2.56	.256	.786	.786E-01	.714E-01	.714E-02	21
22	11	1.84	.167	-.143	-.130E-01	3.33	.303	1.30	.118	.612E-01	.556E-02	22
23	11	2.74	.249	268.	24.4	3.94	.358	2.52	.229	42.7	3.89	23
24	12	-9.98	-.832	815.	67.9	4.98	.415	3.48	.290	236.	19.7	24
25	12	-6.45	-.538	273.	22.8	4.02	.335	2.51	.209	42.0	3.50	25
26	13	-2.26	-.174	-.402	-.309E-01	4.13	.318	1.59	.122	.877E-01	.674E-02	26
27	13	-.205	-.158E-01	.142	.109E-01	3.32	.256	1.02	.783E-01	.667E-01	.513E-02	27
28	14	.382	.273E-01	-.163	-.116E-01	3.42	.244	1.09	.776E-01	.579E-01	.414E-02	28
29	14	.138	.983E-02	.563E-01	.402E-02	3.12	.223	.859	.614E-01	.493E-01	.352E-02	29
30	15	-3.78	-.252	1.60	.106	5.04	.336	1.97	.132	.134	.890E-02	30
31	15	-5.12	-.341	-.182	-.121E-01	4.79	.319	1.74	.116	.837E-01	.558E-02	31
32	16	-.205	-.128E-01	-.403	-.252E-01	3.27	.205	.987	.617E-01	.271E-01	.169E-02	32

INTEGRALS INVOLVING NEGATIVE FREQUENCIES

SUM FREQ SEQ NO	NO IN SCAN (N)	BR	BR/N	BI	BI/N	BIC	BIC/N	BIC**2	(BIC**2)/N	S(W3)* BIC**2	(S(W3)* BIC**2)/N	SUM FREQ SEQ NO
1	31	2.90	.937E-01	-2.03	-.655E-01	7.81	.252	2.73	.882E-01	.201	.649E-02	1
2	30	1.22	.407E-01	-.972	-.324E-01	8.96	.299	3.40	.113	.120	.400E-02	2
3	29	-.176	-.608E-02	.268	.925E-02	8.73	.301	3.29	.114	.126	.433E-02	3
4	28	-.348	-.124E-01	.373	.133E-01	7.53	.269	2.51	.897E-01	.108	.387E-02	4
5	27	-.943	-.349E-01	-.444E-01	-.165E-02	9.01	.334	3.80	.141	.213	.790E-02	5
6	26	-2.42	-.930E-01	-.701	-.270E-01	7.94	.305	2.98	.115	.166	.640E-02	6
7	25	-4.35	-.174	536.	21.5	9.67	.387	6.13	.245	102.	4.08	7
8	24	2.11	.879E-01	.160E 04	66.8	11.4	.474	8.18	.341	549.	22.9	8
9	23	4.44	.193	528.	23.0	9.44	.411	6.37	.277	107.	4.65	9
10	22	2.15	.977E-01	.400	.182E-01	6.41	.291	2.22	.101	.131	.594E-02	10
11	21	-.411	-.196E-01	-.192	-.913E-02	6.54	.311	2.36	.112	.140	.667E-02	11
12	20	-.367	-.183E-01	-.139E-02	-.695E-04	6.69	.335	2.67	.134	.134	.670E-02	12
13	19	.415E-01	.218E-02	.284	.150E-01	4.63	.244	1.31	.690E-01	.789E-01	.415E-02	13
14	18	.468	.260E-01	.190	.105E-01	3.86	.214	.966	.536E-01	.522E-01	.290E-02	14
15	17	2.06	.121	273.	16.1	6.00	.353	3.62	.213	61.4	3.61	15
16	16	-2.99	-.187	804.	50.2	6.95	.434	4.70	.294	315.	19.7	16
17	15	-5.80	-.387	259.	17.3	6.47	.431	4.26	.284	70.5	4.70	17
18	14	.648E-01	.463E-02	-.736	-.526E-01	4.55	.325	1.76	.126	.738E-01	.527E-02	18
19	13	.218	.167E-01	-.368	-.283E-01	3.57	.274	1.20	.921E-01	.693E-01	.533E-02	19
20	12	.284	.237E-01	-.421	-.351E-01	3.35	.279	1.09	.911E-01	.123	.103E-01	20
21	11	-.564E-02	-.513E-03	-.466	-.423E-01	3.23	.293	1.27	.115	.115	.105E-01	21
22	10	.208	.208E-01	-.235	-.235E-01	2.10	.210	.586	.586E-01	.276E-01	.276E-02	22
23	9	-.598	-.665E-01	.969	.108	2.66	.295	1.23	.136	20.8	2.31	23
24	8	-1.27	-.159	-8.98	-1.12	2.46	.308	1.21	.151	82.1	10.3	24
25	7	-.139	-.198E-01	-2.72	-.388	2.06	.294	1.04	.149	17.5	2.49	25
26	6	.511E-02	.852E-03	-.246E-02	-.409E-03	1.36	.226	.374	.623E-01	.206E-01	.344E-02	26
27	5	-.105E-01	-.211E-02	.807E-03	.161E-03	1.22	.244	.332	.663E-01	.217E-01	.435E-02	27
28	4	.470E-02	.118E-02	-.403E-02	-.101E-02	1.02	.254	.305	.762E-01	.162E-01	.406E-02	28
29	3	.779E-02	.260E-02	-.186E-02	-.620E-03	.876	.292	.282	.940E-01	.162E-01	.540E-02	29
30	2	.177E-01	.887E-02	-.299E-03	-.149E-03	.807	.404	.341	.171	.231E-01	.115E-01	30
31	1	.165E-02	.165E-02	-.293E-02	-.293E-02	.217	.217	.470E-01	.470E-01	.226E-02	.226E-02	31

STOP NORMAL PROGRAM COMPLETION

NAME: BIVEC

TYPE: Main Program

PURPOSE: To calculate auto and cross rotary bispectra of two vector series, to write disc files of rotary bicoherences for transmittal to plotting program RBPLOT,[1] and to compute confidence levels for the rotary bicoherences.

MACHINE: Xerox Sigma-7

SOURCE LANGUAGE: FORTRAN IV

DESCRIPTION:

Program BIVEC uses as input an RWDISC[2] file of Fourier coefficients having a particular order to be described below. These coefficients are assumed to result from two real data series per vector series, the real series constituting the scalar components. The coefficients are normally the output of program FOURIER,[3] which has done any necessary prewhitening, subsampling, overlapping, mean and trend removal, and Hanning.

If L is the piece length and P is the number of pieces resulting from program FOURIER, there are $2 \cdot LP$ words input into BIVEC for each vector data series. Program BIVEC computes approximately $(3/2)(L/2)^2$ values of auto rotary bicoherence, or either $(3/2)(L/2)^2$ or $3(L/2)^2$ values of cross rotary bicoherence, the latter depending on the combination of sources of data. A normalization of rotary bicoherences by amplitude can be specified (see below). Corresponding rotary biphase and rotary biphase error are also computed if desired. In normal use, program BIVEC writes consecutive files of rotary bicoherences on the disc to be used as input to plotting program RBPLOT. The user can instruct the program to list rotary bicoherences and related quantities when the rotary bicoherences fall between limits set by the user. Finally, a listing of partitioning of rotary bicoherences into bins of specified size can be obtained, as well as a list of rotary bicoherences which correspond to specified confidence levels.

COMPUTATIONS PERFORMED:

I. Rotary Bispectra

The default definitions of rotary bispectrum, rotary bicoherence, and rotary biphase used by program BIVEC are those given by N. C. G. Yao. (4),(5) From two real series, $u_1(t)$ and $u_2(t)$, construct the complex vector series $\vec{u}(t) = u_1(t) + iu_2(t)$. Let ω be the angular velocity of $\vec{u}(t)$ and let $\sigma = |\omega|$. Further let $A_1(\omega)$, $B_1(\omega)$ be the cosine and sine Fourier coefficients of real series $u_1(t)$ and $A_2(\omega)$, $B_2(\omega)$ be the corresponding quantities for real series $u_2(t)$.

One can represent the vector series $\vec{u}(t)$ by

$$\vec{u}(t) = u_1(t) + iu_2(t) = \sum_{\sigma=0}^{\infty} (U_+(\sigma)e^{i\sigma t} + U_-(\sigma)e^{-i\sigma t})$$

where the coefficients $U_+(\sigma)$ and $U_-(\sigma)$ are called by Yao "rotary Fourier coefficients," and they correspond respectively to positive and negative angular frequencies present in the vector series $\vec{u}(t)$. Yao shows that $U_+(\sigma)$ and $U_-(\sigma)$ are given by:

$$U_+(\sigma) = [A_1(\sigma) + B_2(\sigma)] + i[A_2(\sigma) - B_1(\sigma)]; \quad \omega = \sigma$$

$$U_-(\sigma) = [A_1(\sigma) - B_2(\sigma)] + i[A_2(\sigma) + B_1(\sigma)]; \quad \omega = -\sigma$$

in terms of Fourier coefficients of the component real series.

The rotary spectrum $\langle P(\omega)d\omega\rangle$ is then defined as:

$$\begin{aligned} \langle P(\omega)d\omega\rangle &= \langle U(\omega_1)U^*(\omega_2)\rangle && \text{when } \omega_1 = \omega_2 \\ &= 0 && \text{when } \omega_1 \neq \omega_2 \end{aligned} \qquad (1)$$

and the rotary bispectrum $\langle RB(\omega_1,\omega_2)d\omega^2\rangle$ as

$$\langle RB(\omega_1,\omega_2)d\omega^2\rangle = \langle U(\omega_1)U(\omega_2)U^*(\omega_3)\rangle \quad \text{when } \omega_1 + \omega_2 = \omega_3$$
$$\text{when } \omega_1 + \omega_2 \neq \omega_3 \qquad (2)$$

where $U(\omega)$ represents $U_+(\sigma)$ or $U_-(\sigma)$ depending on the sign of ω. In the largest frequency domain of interest (see below) there will be six possible interpretations of equation (2) in terms of magnitude σ.

The rotary bicoherence Rbic (ω_1,ω_2) is defined as:

$$\mathrm{Rbic}(\omega_1,\omega_2) = \frac{|\langle RB(\omega_2,\omega_2)d\omega^2\rangle|}{[\langle P(\omega_1)\rangle\langle P(\omega_2)\rangle\langle P(\omega_3)\rangle d\omega^3]^{1/2}} \quad (3)$$

and the rotary biphase $R\phi(\omega_1,\omega_2)$ as:

$$R\phi(\omega_1,\omega_2) = \tan^{-1}\frac{\mathrm{Im}\langle RB(\omega_1,\omega_2)d\omega^2\rangle}{\mathrm{Re}\langle RB(\omega_1,\omega_2)d\omega^2\rangle} \quad (4)$$

If P is the number of pieces, the averaging procedure indicated in equations (1) and (2) leads to:

$$\mathrm{Rbic}(\omega_1,\omega_2) = \frac{P^{1/2}\left|\sum_{k=1}^{P} RB_k(\omega_1,\omega_2)d\omega^2\right|}{\left[\sum_{k=1}^{P} P_k(\omega_1)d\omega \sum_{k=1}^{P} P_k(\omega_2)d\omega \sum_{k=1}^{P} P_k(\omega_3)d\omega\right]^{1/2}} \quad (5)$$

where k identifies each piece.

Alternate form of rotary bicoherence

Program BIVEC can compute an alternate form of rotary bicoherence based on the following definition of rotary bispectrum normalized by its amplitude:

$$\langle RB'(\omega_1,\omega_2)d\omega^2\rangle = \left\langle \frac{RB_k(\omega_1,\omega_2)d\omega^2}{|RB_k(\omega_1,\omega_2)d\omega^2|}\right\rangle \quad (6)$$

where $RB_k(\omega_1,\omega_2)d\omega^2$ is given by equation (2) applied to piece k.

Using this definition, the piece-averaged rotary bicoherence is now:

$$Rbic'(\omega_1,\omega_2) = |\langle RB'(\omega_1,\omega_2)d\omega^2\rangle| \qquad (7)$$

$$= \frac{1}{P}\left|\sum_{k=1}^{P} \frac{RB_k(\omega_1,\omega_2)d\omega^2}{|RB_k(\omega_1,\omega_2)d\omega^2|}\right|$$

Frequency domain

As stated above, program BIVEC can operate on one or two vector series. Let the first be represented by X and the second by Y. Let the juxtaposition of three symbols from the set {X,Y} refer to a particular choice of sources for Fourier coefficients of frequency $\omega_1,\omega_2,\omega_3$ respectively. It can be shown that there are only four distinct rotary bispectra that can be calculated from the two vector series X and Y:

1) auto rotary bispectrum for XXX
2) auto rotary bispectrum for YYY
3) cross rotary bispectrum for XXY
4) cross rotary bispectrum for XYY

It can further be shown that it is not necessary to let the constituent frequencies range over all possible values. The minimum necessary domain for cases 1), 2), 3) is:

$$\{A\} = \{(|\omega_1| \geq |\omega_2|) \cap (-L/2 \leq \omega_1 < L/2) \cap$$
$$(-\omega_1 - L/2 \leq \omega_2 \leq -\omega_1 + L/2) \cap (\omega_2 \neq 0) \cap (\omega_2 \neq -\omega_1)\}$$

where $\omega_3 = \omega_1 + \omega_2$, L is the piece length, and ω is expressed in units of cycles per point. Pictorially, this is the unshaded area in figure 1. For case (4), however, it is necessary to approximately double this domain so that the shaded area in figure 1 is also included. The minimum necessary domain is now:

$$\{A'\} = \{(-L/2 \leq \omega_1 \leq L/2) \cap (-L/2 \leq \omega_2 \leq L/2) \cap$$
$$(-\omega_1 - L/2 \leq \omega_2 \leq -\omega_1 + L/2) \cap (\omega_2 \neq 0) \cap (\omega_2 \neq -\omega_1)\}$$

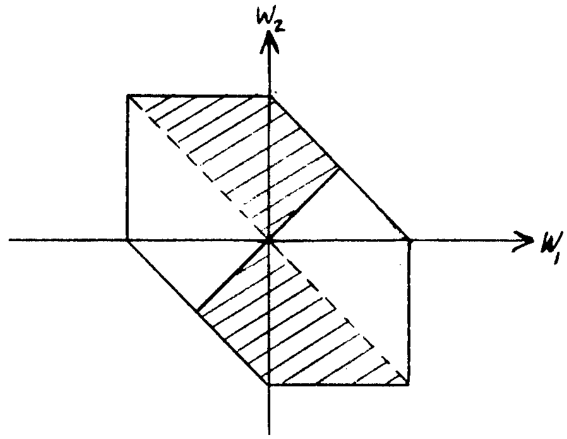


Figure 1

Order of the computations

Program BIVEC can handle two vector data series, each consisting of two scalar series each of length 16 K words. This means that for 50% overlap and for the limiting case of an infinite number of pieces, each vector series entering the program consists of 64K Fourier coefficients. In order to permit computation of the bispectra, and to minimize the number of disc accesses, the following steps are taken:

1) Organization of input file

Let $\{a_i\}$ be the set of all real parts of the i^{th}-frequency Fourier coefficient taken over all pieces, and let $\{b_i\}$ be the corresponding imaginary parts. Then, for a piece length of L, the input file elements have the following order:

$$\{a_o\}, \{a_1\}, \{b_1\}, \{a_2\}, \{b_2\}, \ldots \{a_{\frac{L}{2}-1}\},$$

$$\{b_{\frac{L}{2}-1}\}, \{a_{\frac{L}{2}}\} .$$

The elements of $b_{\frac{L}{2}}$ are all zero and do not appear in the input file. Program FOURIER (3) automatically outputs coefficients in this order. This permits all elements associated with a given frequency to be input by entering the series only once.

2) Organization of frequency domain and computation procedure

Figure 2 is a representation of 1/4 of the maximum domain described above.

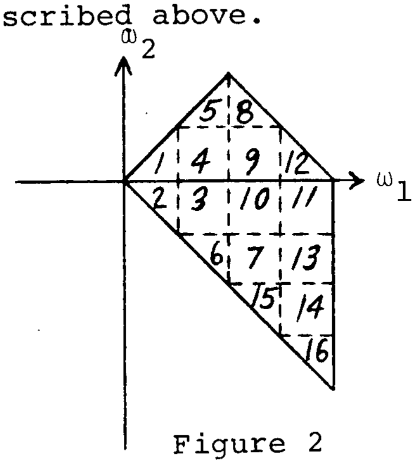


Figure 2

This area is shown divided into 16 sub-domains or "rasters." Program BIVEC calculates rotary bispectra for each raster in the order of numbering in fig. 2. Simultaneous with this calculation, is computation of rotary bispectra for the raster which is symmetric about the origin, having sum frequency w_3 of opposite sign, and which uses the same set of Fourier coefficients. In this way, there are no more than LP/4 elements from each component scalar series in core at any time, where L = piece length, P = number of pieces. The reading of series "quarters" from the disc is a function of each particular combination of source series, an attempt being made to exploit symmetries for maximum efficiency. Within each raster, computations are done from upper left to lower right along diagonal scans of slope (-1), each scan thus corresponding to a single sum frequency ω_3. The disc is accessed for the coefficients of frequency ω_3 only when these coefficients are not already available in core.

Computations over the shaded area of fig. 1 are done similarly.

II. Confidence Levels

As each rotary bicoherence is computed, the user may elect to assign it to a bin having a size of his choice. The user may also determine the number of bins by specifying

the upper rotary bicoherence limit to be considered. After this sorting has been done, program BIVEC determines the rotary bicoherences below which specified fractions ("confidence levels") of the total number or rotary bicoherences calculated by the program lie. This is done by calculating the fraction of the total which lies below the upper limit of each bin, finding the two closest such fractions which "bracket" the specified confidence level, and linearly interpolating the two corresponding rotary bicoherences to find the one which corresponds to the specified confidence level. A dump of the distribution is available.

USAGE:

(1) Tailoring load module size

For long series of Fourier coefficients, it is desirable to have large data buffers to keep disc accesses to a minimum. On the other hand these large data buffers can be wasteful for shorter series both in terms of (page-minutes) charges and the penalty in turnaround time. The user is therefore advised to consider recompiling the program to match his needs if he is going to be handling a number of data series of comparable length. To edit the source program prior to recompilation, the following steps should be observed:

1. Copy the source file to a duplicate file, and then work with the duplicate file;
2. Change the DIMENSIONs of W11, W12, W21, W22 to at least LP/4, where L = piece length and P is the number of pieces.
3. Make the first dimension of OM exactly the same as in (2).
4. Check that arrays W31 and W32 are DIMENSIONed at least 2P.
5. Check that SQ1 and SQ2 are DIMENSIONed at least (L/8, L/8), both arrays identically.
6. Check that the initialization of variable ISQD is identical to either dimension in (5), and that ISOM is initialized at exactly the dimensions in (2).
7. Recompile.

(2) Control Cards

Let aaa = account number
uuu = user number
fff = input file name (RWDISC, keyed)
ttii = file names for transmittal to plotting program (consecutive files)

Assume the element file is called BIVECR and exists in account aaa. The Control Card portion of the input deck is then as follows:

```
     !JOB aaa,uuu
   * !LIMIT (TIME,10),(CORE,30)
     !ASSIGN F:RAD, (FILE,fff),  (DIRECT),(KEYED)
   † !ASSIGN F:1, (FILE,tt11), (OUT),(SAVE)
   † !ASSIGN F:2, (FILE,tt22), (OUT),(SAVE)
  †† !ASSIGN F:3, (FILE,tt33), (OUT),(SAVE)
  †† !ASSIGN F:4, (FILE,tt44), (OUT),(SAVE)
     !LOAD (EF,(BIVECR)), (UNSAT,(3))
     !RUN
     !DATA
     (DATA CARDS)
```

(3) Data cards

Program BIVEC has NAMELIST input followed by one or two data cards depending on the application. The NAMELIST input must be terminated by an * card whether or not any NAMELIST variables are being input.

The NAMELIST parameters and their default values are as follows. In general the switches are to be set to 0 for "off" and 1 for "on".

---

*Time and core limits will vary widely depending on application. See below.

†Necessary only when storing bicoherences on disc, for transmittal to plotting program. DCB assignments may be changed by NAMELIST parameters ISTORE1 and ISTORE2 respectively.

††Must be included along with DCB assignments mentioned in footnote †, when doing cross rotary bispectrum for case XYY and transmitting bicoherences via disc to plotting program. May be changed by NAMELIST parameters ISTORE3 and ISTORE4 respectively.

NAMELIST VARIABLE	MEANING	DEFAULT VALUE
MF	Number of logical files in input RAD file	21
IDATSTART	Starting location of data in each input logical file	5
LISTBI	Switch to list rotary bicoherences and associated quantities	0
SIGLVL	95% confidence level for rotary bicoherences, above which phase error PHSERR is calculated and printed when LISTBI=1	*
IDUMP	Switch to store rotary bicoherences in a consecutive file for transmittal to plotting program	1
ISTORE1 ISTORE2	DCB assignments for consecutive files of rotary bicoherences created by setting switch IDUMP. Applies always	1 2
ISTORE3 ISTORE4	Like ISTORE1 and ISTORE2, except that these apply only for cross rotary bispectrum in the case XYY	3 4
INORM	Sets amplitude normalization described above	0
LABREAD	Switch to read labels for each series in input RWDISC file and to write these labels to output files for use by the plotting program. Each label consists of original buoy file name and variable number. If not set, then EBCDIC blanks will be written to output file	1

---

*Default is an approximate value. If P = number of pieces, let the number of degrees of freedom EDOF be defined as follows:

$$\text{EDOF} = \begin{cases} 36P^2/(19P-1) & \text{for Hanning, overlapping} \\ 2P & \text{for no Hanning, no overlapping} \end{cases}$$

Then the default value of SIGLVL is

$$\text{SIGLVL} = (6/\text{EDOF})^{1/2}$$

NAMELIST VARIABLE	MEANING	DEFAULT VALUE
IFOURIER	Switch to read 132 character label in input file describing processing done by program FOURIER and to write these labels to output files for use by the plotting program. If not set, then EBCDIC blanks will be written to output file	1
KONF	Switch for determining rotary bicoherence confidence levels	0
IHANN	Determines method of calculating phase error when LISTBI≠0 (See PHSERR, page 13) IHANN=0 not Hanned and not overlapped IHANN=1 Hanned and overlapped	1
BINSIZE	Bin width for sorting rotary bicoherences. A given rotary bicoherence is placed in bin I, where I = INT (rot. bicoh./ BINSIZE) + 1	0.01
RBICLIM	The rotary bicoherence "limit" for sorting which determines the number of bins of size BINSIZE, by determining the integer NBINS obtained by rounding (RBICLIM/BINSIZE), and then redefining RBICLIM as (NBINS)X(BINSIZE)	1.5
KB1	Switch to display partitioning of rotary bicoherences	1
KB2	Switch to display fraction of total rotary bicoherences under each bin limit	1

The data cards follow the * card which terminates NAMELIST input. They are all in generalized, free-field format and are as follows:

Card 1 (Mandatory)

Variable No.	Meaning
* 1	Logical file for 1st scalar component of 1st vector series
* 2	Logical file for 2nd scalar component of 1st vector series
* 3	Logical file for 1st scalar component of 2nd vector series
* 4	Logical file for 2nd scalar component of 2nd vector series
* 5	Logical file of 1st scalar component for sum frequency series
* 6	Logical file of 2nd scalar component for sum frequency series
7	Total number of pieces in each series
** 8	Piece length
9, 10	Lower, upper piece sequence numbers inclusive, between which user wants to average
11, 12	Lower, upper limits determining when to print rotary bicoherences when LISTBI$\neq$0. Rotary bicoherences must satisfy: (var. #11) $\leq$ rotary bicoherence $\leq$ (var. #12)

Card 2 (Input only when KONF$\neq$0)

Variable No.	Meaning
1 (NCONF)	Number of confidence levels to follow ($\leq$10)
2 (CONF)	NCONF confidence levels expressed as decimal, 0 to 1

---

*The user must take care to use no more than two distinct vector series or the program will not proceed. Two vector series are distinct unless both 1st and 2nd scalar component series are respectively identical. The program automatically determines the form of rotary bispectrum discussed under "Frequency Domain," as well as the corresponding frequency domain.

**Piece length must be evenly divisible by 8, have no prime factor greater than 5, and be no greater than 4000. Appendix 2 is a list of permissible choices.

OUTPUT:

(1) Line Printer

Typical line printer output is displayed in Appendix 1. First a dump of NAMELIST parameter values is printed, followed by a facsimile of input data cards. Next, if confidence levels are called for, the "adjusted rotary bicoherence limit" (as determined by the rule given in describing variable RBICLIM in the section on Data Cards above) is printed.

This is followed by a listing of individual rotary bicoherences if called for. There is one rotary bicoherence value per line, plus other output described below. The listing occurs only within the limits given by variables #11 and #12 on Card 1, and the order is the order of computation. Every positive sum frequency considered is followed by the corresponding negative sum frequency. The user is cautioned that for cross rotary bicoherences of form XYY, the total number of rotary bicoherences computed is about $3(L/2)^2$, which is nearly 50000 for a piece length of 256. Without sufficiently narrow limits on the rotary bicoherences actually printed, this can result in a prohibitively large number of output pages.

The items output in each line depend on which form of rotary bicoherence is called for in the program. They are as follows, in order of appearance from left to right on the line:

(A) Default rotary bicoherence (after Yao):

Variable	Meaning
F1 F2 F3	Frequency triplet (sequence numbers) associated with the rotary bicoherence
RBIC	Rotary bicoherence
RBPH	Rotary biphase
RNUM	$\lvert\langle RB(\omega_1,\omega_2)d\omega^2\rangle\rvert$
RBR	$\langle \text{Real } [RB(\omega_1,\omega_2)d\omega^2]\rangle$
RBI	$\langle \text{Imag } [RB(\omega_1,\omega_2)d\omega^2]\rangle$
P1	$\langle P(\omega_1)d\omega\rangle$
P2	$\langle P(\omega_2)d\omega\rangle$
P3	$\langle P(\omega_3)d\omega\rangle$

Variable	Meaning
RDEN	$[\langle P(\omega_1)d\omega\rangle\langle P(\omega_2)d\omega\rangle\langle P(\omega_3)d\omega\rangle]^{1/2}$
PHSERR*	Phase error; depends on IHANN (see page 9) $= \sqrt{2P}$ for IHANN=0, P = No. of pieces $= 57.296 \sin^{-1} [1.96/(\mathrm{RBIC} \cdot W)]$ for IHANN=1 default where $W = [36P^2/(19P-1)]^{1/2}$

(B) Alternate rotary bicoherence (normalized by amplitude)

Variable	Meaning
F1 F2 F3	Frequency triplet (sequence numbers) associated with this rotary bicoherence
RBIC	Rotary bicoherence
RBPH	Rotary biphase
RBR	$\langle \mathrm{Real}\ [RB'(\omega_1,\omega_2)d\omega^2]\rangle$
RBI	$\langle \mathrm{Imag}\ [RB'(\omega_1,\omega_2)d\omega^2]\rangle$
PHSERR	Defined as above

After listing individual rotary bicoherences, there is an optional listing of how the rotary bicoherences are partitioned among bins, followed by the total number of rotary bicoherences submitted. The format is displayed in Appendix 1.

Following this there is an optional display of cumulative fractional numbers of rotary bicoherences lying below each upper bin limit. Again, the format is displayed in Appendix 1.

Finally, when a calculation of confidence levels is called for, a summary of the result for each confidence level is displayed as in Appendix 1. This is followed by a statement of the number of rotary bicoherences which happened to fall above the limit used in the program (also stated).

(2) Disc

Normally two or four consecutive disc files are produced by program BIVEC, as explained above, which are used to transmit the computed rotary bicoherences to the plotting programs. These files also contain information about the origin of the data series, the processing history, and the organization of the input file of Fourier coefficients.

---

*PHSERR is calculated and printed only above 95% confidence level given by SIGLVL (see NAMELIST variable table). In the line printer output shown in Appendix 1, SIGLVL=0.5612.

Each output consecutive disc file is organized as follows:

Words, inclusive	Number of Words	Meaning
1 thru N	N	N rotary bicoherences
N+1 thru N+3	3	Buoy format file name for source of variable 1, Card 1
N+4	1	Buoy format variable number for source of variable 1, Card 1
N+5 thru N+24	20	Like words N+1 thru N+4, for variables 2 thru 6, Card 1
N+25	1	No. of logical files in input RWDISC file (usually 21)
N+26 thru N+58	33	Processing history from program FOURIER

The first task of the plotting program is to construct from each consecutive file of rotary bicoherences a two-dimensional array whose elements occupy the same relative position in the matrix as do the grid points in the final plot. By using the coding which does this the user can access this file directly. The author can be consulted for details.

RESTRICTIONS:

1. The length of the data series which program BIVEC can handle depends on the dimensions of arrays W11, W12, W21, W22 (see "USAGE"), as well as on how the data series was split when processed. There is an upper limit of 16K words on the data series length, or 32K words on the length of the resulting Fourier coefficient series entering BIVEC. By "data series" is meant each component series of each vector series, so that 128K Fourier coefficients can actually be used.

   Let

   L = maximum length of data series. 2*L* is assumed to be an exact multiple of L, the piece length, for 50% overlap and *L* is assumed to be an exact multiple of L for no overlap.

   L = piece length

   D = dimensions of the arrays WIJ

   P = no. of pieces

50% overlap

$$\mathcal{L} = (4D+L)/2 \quad \text{and} \quad P = 2\mathcal{L}/L-1 \qquad (8)$$

No overlap

$$\mathcal{L} = 4D \quad \text{and} \quad P = \mathcal{L}/L \qquad (9)$$

In all cases the number of points in the Fourier coefficient series entering BIVEC is PL, and the four WIJ arrays mentioned above must be dimensioned at least PL/4.

2. The first dimension of OM must be exactly the dimension of WIJ arrays mentioned in (1).

3. The arrays W31 and W32 must be dimensioned at least 2P, where P = no. of pieces.

4. The arrays SQI, I=1,2 must be dimensioned at least SQI(M,M) where M = L/8, L = piece length.

5. Variable ISQD must be initialized to either dimension of the arrays SQI(M,M) mentioned in (4). Variable ISOM must be initialized at exactly the dimension of the arrays WIJ mentioned in (1).

6. L must be evenly divisible by 8, have no prime factor greater than 5, and be $\leq$4000.

7. (No. of confidence levels) $\leq$10.

8. (No. of bins used in sorting rotary bicoherences) $\leq$200

ERRORS AND DIAGNOSTICS:

Message	Meaning	Action Taken
"LAST PIECE TO AVG. GREATER THAN NO OF PCS"	Self-explanatory	Abort
"NO MORE THAN TWO DIFFERENT VECTOR SERIES ALLOWED"	(Based on choice of logical files in input RWDISC file). Vector series are different unless both component series are the same	Abort
"TWO OUTPUT DCB's HAVE BEEN SET EQUAL CAN'T DO"	Self-explanatory	Abort

"PIECE LENGTH MUST BE FROM FOLLOWING LIST," followed by a list of acceptable values through 4000	Piece length fails to satisfy one of following: 1) evenly divisible by 8 2) no prime factor greater than 5 3) ≤ 4000	Abort
"NUMBER OF BINS MUST NOT EXCEED 200"	Self-explanatory	Abort

STORAGE AND CPU TIME REQUIREMENTS:

As of the date of this report, program BIVEC has not been run extensively on long series. An operating history would be useful to the user to enable him to tailor the size of the program. It is intended to add to the following list as more runs of the program are reported.

TASK	PIECE LENGTH	NO. OF PIECES	DIMENSIONS OF W11,W12,W21,W22	DIMENSIONS OF W31,W32	DIMENSIONS OF SQ1,SQ2	PEAK CORE (PAGE SIZE 512)	CPU TIME (MIN)	CHARGE (CU)
AUTO ROTARY BICOH. SORTING	256	39	2560	100	(35,35)	50	9.38	15.85
AUTO ROTARY BICOH.	256	39	2560	100	(35,35)	50	9.34	15.56

The sorting above was done to determine confidence levels from Gaussian noise. In both cases the same number of rotary bicoherences was calculated. It is anticipated that calculation of cross rotary bicoherences for the case XXY on the above series would take essentially the same time, and for the case XYY would take approximately double the time (and cost), since twice as many rotary bicoherences are calculated.

<u>SUBPROGRAMS REQUIRED:</u> None

<u>PROGRAMMER:</u> Gerard H. Martineau

<u>ORIGINATOR:</u> Melbourne G. Briscoe

<u>DATE:</u> October, 1977

<u>REFERENCES:</u>

(1) Report on Program RBPLOT, by G. H. Martineau.

(2) "Handbook for Computer Users," Information Processing Center of W.H.O.I., pp. V-E-1 ff.

(3) Report on Program FOURIER, by G. H. Martineau.

(4) "Bispectral and Cross-Bispectral Analysis of Wind and Currents off Oregon Coast," Ph.D. Thesis at Oregon State University, by N. C. G. Yao, June, 1974.

(5) "Rotary Cross-Bispectra and Energy Transfer Functions between Non-Gaussian Vector Processes I. Development and Example," by N.-C. Yao, Steve Neshyba, and Henry Crew, <u>Journal of Physical Oceanography</u>, Vol. 5, January, 1975, p. 164.

APPENDIX 1
(BIVEC)

```
17:45 JAN 26,'78 ID=04F4
JOB 462,1719
LIMIT (TIME,3),(CORE,15),(ORDER),(ACCOUNT)
ASSIGN F:RAD,(FILE,BVEC11),(DIRECT),(KEYED)
LOAD (EF,(BIVEC05R)),(UNSAT,(3))
!P1 ASSOCIATED.


* * ALLOCATION SUMMARY * *

  PROTECTION     LOCATION   PAGES

DATA (00)        A000       7
PROCEDURE (01)   B000       3
DCB (10)         AE00       1
******************************   SGN= D4L          SIZE=008.1K   ******************************

 ******** PROTECTION TYPES:  00 DATA       01 PROCEDURE   10 STATIC

                            SEGHI=0  AC61  SEGHI=1  C55D  SEGHI=2  AFFF
                            SEGLO=0  A000  SEGLO=1  B000  SEGLO=2  AE00

                            00 SIZE=  C62  01 SIZE= 155E  10 SIZE=  200
RUN
 IDATSTART = 5
 MF = 21
 ISTORE1 = 1
 ISTORE2 = 2
 ISTORE3 = 3
 ISTORE4 = 4
 BINSIZE = 1.000000E-02
 RBICLIM = 1.50000
 IHI = 1
 KB1 = 1
 KB2 = 1
 KONF = 1
 LISTBI = 1
 IDUMP = 0
 LABREAD = 1
 IFOURIER = 1
 INORM = 0
 IHANN = 1
 1  2  1  2  1  2  10  64  1  10  .545  .575
 4  .90  .95  .99  .999
ADJUSTED ROT BICOH LIMIT, HOLDING BINSIZE AS INPUT   1.50
```

F1	F2	F3	RBICOH	RBIPHASE	/<RB>/	<REAL(RB)>	<IMAG(RB)>	<P(W1)DW>	<P(W2)DW>	<P(W3)DW>	SQRT(PROD<P>)	PHSERR
8	2	10	.566	-25	.166E-01	.150E-01	-.707E-02	.106E 00	.673E-01	.121E 00	.293E-01	52.5
7	5	12	.551	-33	.321E-01	.270E-01	-.173E-01	.144E 00	.155E 00	.152E 00	.583E-01	
-9	7	-2	.546	-99	.240E-01	-.380E-02	-.237E-01	.108E 00	.144E 00	.124E 00	.440E-01	
14	-2	12	.563	-156	.282E-01	-.257E-01	-.114E-01	.133E 00	.124E 00	.152E 00	.501E-01	53.0
13	3	16	.550	-49	.270E-01	.177E-01	-.204E-01	.148E 00	.132E 00	.124E 00	.491E-01	
-14	-2	-16	.569	173	.161E-01	-.160E-01	.197E-02	.706E-01	.124E 00	.917E-01	.284E-01	52.1
11	-10	1	.559	-158	.324E-01	-.300E-01	-.122E-01	.117E 00	.970E-01	.295E 00	.579E-01	
-17	16	-1	.549	-156	.202E-01	-.185E-01	-.823E-02	.586E-01	.124E 00	.187E 00	.368E-01	
-19	9	-10	.551	-83	.187E-01	.232E-02	-.185E-01	.705E-01	.168E 00	.970E-01	.339E-01	
-21	10	-11	.552	-172	.233E-01	-.231E-01	-.344E-02	.110E 00	.121E 00	.134E 00	.423E-01	
22	3	25	.561	81	.174E-01	.283E-02	.172E-01	.639E-01	.132E 00	.115E 00	.311E-01	

F1	F2	F3	RBICOH	RBIPHASE	/<RB>/	<REAL(RB)>	<IMAG(RB)>	<P(W1)DW>	<P(W2)DW>	<P(W3)DW>	SQRT(PROD<P>)	PHSERR
-23	-5	-28	.574	28	.217E-01	.193E-01	.101E-01	.136E 00	.995E-01	.106E 00	.378E-01	51.5
20	-3	17	.554	-143	.238E-01	-.231E-01	-.173E-01	.219E 00	.121E 00	.102E 00	.521E-01	
-22	1	-21	.569	-15	.369E-01	.357E-01	-.932E-02	.129E 00	.295E 00	.110E 00	.649E-01	52.1
26	-6	20	.558	61	.283E-01	.137E-01	.247E-01	.162E 00	.722E-01	.219E 00	.507E-01	
27	-6	21	.558	86	.235E-01	.166E-02	.234E-01	.129E 00	.722E-01	.191E 00	.421E-01	
-30	2	-28	.566	4	.159E-01	.159E-01	.115E-02	.111E 00	.673E-01	.106E 00	.281E-01	52.5
32	-1	31	.547	95	.258E-01	-.211E-02	.258E-01	.574E-01	.187E 00	.208E 00	.472E-01	
-28	-1	-29	.546	-157	.225E-01	-.207E-01	-.868E-02	.106E 00	.187E 00	.855E-01	.411E-01	
-25	12	-13	.563	28	.136E-01	.120E-01	.630E-02	.638E-01	.152E 00	.599E-01	.241E-01	52.8
-26	9	-17	.559	-67	.180E-01	.708E-02	-.166E-01	.106E 00	.168E 00	.586E-01	.323E-01	
-30	11	-19	.572	-179	.173E-01	-.173E-01	-.315E-03	.111E 00	.117E 00	.705E-01	.302E-01	51.8
30	-9	21	.572	142	.239E-01	-.189E-01	.147E-01	.847E-01	.108E 00	.191E 00	.418E-01	51.7
28	-19	9	.554	-159	.169E-01	-.158E-01	-.597E-02	.785E-01	.705E-01	.168E 00	.305E-01	
-29	19	-10	.557	33	.197E-01	.164E-01	.108E-01	.855E-01	.150E 00	.970E-01	.353E-01	
-30	20	-10	.553	101	.268E-01	-.520E-02	.263E-01	.111E 00	.219E 00	.970E-01	.486E-01	
31	-29	2	.558	-7	.193E-01	.192E-01	-.240E-02	.208E 00	.855E-01	.673E-01	.346E-01	

BIN NO	BIN LIMITS		NO OF ROT BIC'S
1	.00	.01	3
2	.01	.02	4
3	.02	.03	9
4	.03	.04	7
5	.04	.05	19
6	.05	.06	10
7	.06	.07	11
8	.07	.08	22
9	.08	.09	16
10	.09	.10	35
11	.10	.11	30
12	.11	.12	35
13	.12	.13	28
14	.13	.14	28
15	.14	.15	40
16	.15	.16	29
17	.16	.17	40
18	.17	.18	35
19	.18	.19	35
20	.19	.20	40
21	.20	.21	47
22	.21	.22	36
23	.22	.23	37
24	.23	.24	47
25	.24	.25	35
26	.25	.26	39
27	.26	.27	29
28	.27	.28	26
29	.28	.29	42
30	.29	.30	47
31	.30	.31	36
32	.31	.32	40
33	.32	.33	37
34	.33	.34	31
35	.34	.35	32
36	.35	.36	35
37	.36	.37	23
38	.37	.38	34
39	.38	.39	32
40	.39	.40	32
41	.40	.41	20

42	.41	.42	22
43	.42	.43	25
44	.43	.44	27
45	.44	.45	20
46	.45	.46	22
47	.46	.47	12
48	.47	.48	12
49	.48	.49	11
50	.49	.50	6
51	.50	.51	7
52	.51	.52	9
53	.52	.53	14
54	.53	.54	5
55	.54	.55	7
56	.55	.56	13
57	.56	.57	7
58	.57	.58	6
59	.58	.59	9
60	.59	.60	11
61	.60	.61	5
62	.61	.62	2
63	.62	.63	3
64	.63	.64	3
65	.64	.65	2
66	.65	.66	2
67	.66	.67	3
68	.67	.68	4
69	.68	.69	3
70	.69	.70	1
71	.70	.71	2
72	.71	.72	2
73	.72	.73	1
74	.73	.74	0
75	.74	.75	5
76	.75	.76	2
77	.76	.77	0
78	.77	.78	0
79	.78	.79	2
80	.79	.80	2
81	.80	.81	0
82	.81	.82	0
83	.82	.83	0
84	.83	.84	0
85	.84	.85	0
86	.85	.86	0
87	.86	.87	0
88	.87	.88	0
89	.88	.89	1
90	.89	.90	0
91	.90	.91	0
92	.91	.92	0
93	.92	.93	0
94	.93	.94	0
95	.94	.95	0
96	.95	.96	0
97	.96	.97	0
98	.97	.98	0
99	.98	.99	0
100	.99	1.00	0
101	1.00	1.01	0

BIN NØ	BIN LIMITS		NØ ØF RØT BIC'S
102	1.01	1.02	0
103	1.02	1.03	0
104	1.03	1.04	0
105	1.04	1.05	0
106	1.05	1.06	0
107	1.06	1.07	0
108	1.07	1.08	0
109	1.08	1.09	0
110	1.09	1.10	0
111	1.10	1.11	1
112	1.11	1.12	0
113	1.12	1.13	0
114	1.13	1.14	0
115	1.14	1.15	0
116	1.15	1.16	0
117	1.16	1.17	0
118	1.17	1.18	0
119	1.18	1.19	0
120	1.19	1.20	0
121	1.20	1.21	0
122	1.21	1.22	0
123	1.22	1.23	0
124	1.23	1.24	0
125	1.24	1.25	0
126	1.25	1.26	0
127	1.26	1.27	0
128	1.27	1.28	0
129	1.28	1.29	0
130	1.29	1.30	0
131	1.30	1.31	0
132	1.31	1.32	0
133	1.32	1.33	0
134	1.33	1.34	0
135	1.34	1.35	0
136	1.35	1.36	0
137	1.36	1.37	0
138	1.37	1.38	0
139	1.38	1.39	0
140	1.39	1.40	0
141	1.40	1.41	0
142	1.41	1.42	0
143	1.42	1.43	0
144	1.43	1.44	0
145	1.44	1.45	0
146	1.45	1.46	0
147	1.46	1.47	0
148	1.47	1.48	0
149	1.48	1.49	0
150	1.49	1.50	0

TØTAL NØ ØF RØT BIC'S SUBMITTED: 1504

BIN NØ	BIN LIMITS		FRACT BELØW UPPER BIN LIM
1	.00	.01	.002
2	.01	.02	.005
3	.02	.03	.011
4	.03	.04	.015

5	.04	.05	.028
6	.05	.06	.035
7	.06	.07	.042
8	.07	.08	.057
9	.08	.09	.067
10	.09	.10	.090
11	.10	.11	.110
12	.11	.12	.134
13	.12	.13	.152
14	.13	.14	.171
15	.14	.15	.197
16	.15	.16	.217
17	.16	.17	.243
18	.17	.18	.267
19	.18	.19	.290
20	.19	.20	.316
21	.20	.21	.348
22	.21	.22	.372
23	.22	.23	.396
24	.23	.24	.428
25	.24	.25	.451
26	.25	.26	.477
27	.26	.27	.496
28	.27	.28	.513
29	.28	.29	.541
30	.29	.30	.572
31	.30	.31	.596
32	.31	.32	.623
33	.32	.33	.648
34	.33	.34	.668
35	.34	.35	.689
36	.35	.36	.713
37	.36	.37	.728
38	.37	.38	.751
39	.38	.39	.772
40	.39	.40	.793
41	.40	.41	.807
42	.41	.42	.821
43	.42	.43	.838
44	.43	.44	.856
45	.44	.45	.869
46	.45	.46	.884
47	.46	.47	.892
48	.47	.48	.900
49	.48	.49	.907
50	.49	.50	.911
51	.50	.51	.916
52	.51	.52	.922
53	.52	.53	.931
54	.53	.54	.934
55	.54	.55	.939
56	.55	.56	.947
57	.56	.57	.952
58	.57	.58	.956
59	.58	.59	.962
60	.59	.60	.969
61	.60	.61	.973
62	.61	.62	.974
63	.62	.63	.976
64	.63	.64	.978

BIN NO	BIN LIMITS		FRACT BELOW UPPER BIN LIM
65	.64	.65	.979
66	.65	.66	.981
67	.66	.67	.983
68	.67	.68	.985
69	.68	.69	.987
70	.69	.70	.988
71	.70	.71	.989
72	.71	.72	.991
73	.72	.73	.991
74	.73	.74	.991
75	.74	.75	.995
76	.75	.76	.996
77	.76	.77	.996
78	.77	.78	.996
79	.78	.79	.997
80	.79	.80	.999
81	.80	.81	.999
82	.81	.82	.999
83	.82	.83	.999
84	.83	.84	.999
85	.84	.85	.999
86	.85	.86	.999
87	.86	.87	.999
88	.87	.88	.999
89	.88	.89	.999
90	.89	.90	.999
91	.90	.91	.999
92	.91	.92	.999
93	.92	.93	.999
94	.93	.94	.999
95	.94	.95	.999
96	.95	.96	.999
97	.96	.97	.999
98	.97	.98	.999
99	.98	.99	.999
100	.99	1.00	.999
101	1.00	1.01	.999
102	1.01	1.02	.999
103	1.02	1.03	.999
104	1.03	1.04	.999
105	1.04	1.05	.999
106	1.05	1.06	.999
107	1.06	1.07	.999
108	1.07	1.08	.999
109	1.08	1.09	.999
110	1.09	1.10	.999
111	1.10	1.11	1.000
112	1.11	1.12	1.000
113	1.12	1.13	1.000
114	1.13	1.14	1.000
115	1.14	1.15	1.000
116	1.15	1.16	1.000
117	1.16	1.17	1.000
118	1.17	1.18	1.000
119	1.18	1.19	1.000
120	1.19	1.20	1.000
121	1.20	1.21	1.000
122	1.21	1.22	1.000
123	1.22	1.23	1.000
124	1.23	1.24	1.000

125	1.24	1.25	1.000
126	1.25	1.26	1.000
127	1.26	1.27	1.000
128	1.27	1.28	1.000
129	1.28	1.29	1.000
130	1.29	1.30	1.000
131	1.30	1.31	1.000
132	1.31	1.32	1.000
133	1.32	1.33	1.000
134	1.33	1.34	1.000
135	1.34	1.35	1.000
136	1.35	1.36	1.000
137	1.36	1.37	1.000
138	1.37	1.38	1.000
139	1.38	1.39	1.000
140	1.39	1.40	1.000
141	1.40	1.41	1.000
142	1.41	1.42	1.000
143	1.42	1.43	1.000
144	1.43	1.44	1.000
145	1.44	1.45	1.000
146	1.45	1.46	1.000
147	1.46	1.47	1.000
148	1.47	1.48	1.000
149	1.48	1.49	1.000
150	1.49	1.50	1.000

CONFIDENCE LEVEL (PERCENT)	ROTARY BICOHERENCE
90.0	.481
95.0	.565
99.0	.715
99.9	.885

NO. OF ROT BIC'S GREATER THAN SPECIFIED MAXIMUM OF 1.50 IS 0

STOP NORMAL PROGRAM COMPLETION

APPENDIX 2
(BIVEC)

Appendix 2

Permissible Piece Lengths for Program BIVEC

16	384	1440	3456
24	400	1536	3600
32	432	1600	3848
40	480	1728	3888
48	512	1800	4000
64	576	1920	
72	600	1944	
80	640	2000	
96	648	2048	
120	720	2160	
128	768	2304	
144	800	2400	
160	864	2560	
200	960	2592	
216	1024	2880	
240	1080	2916	
256	1152	3000	
288	1200	3072	
320	1280	3200	
360	1296	3240	

NAME: RBPLOT

TYPE: Main Program

PURPOSE: To construct auto or cross rotary bicoherence contour plots from disc files produced by program BIVEC[1]

MACHINE: Xerox Sigma-7

SOURCE LANGUAGE: FORTRAN IV

DESCRIPTION:

Program RBPLOT reads consecutive disc files of auto or cross rotary bicoherences produced by program BIVEC and then makes a contour plot of these rotary bicoherences over an appropriate domain. The program presently exists in two versions, RBPLOT05 which makes separately annotated plots (in separate plot files) each over 1/4 of the maximum domain as explained below, and RBPLOT06 which makes a single plot over the entire domain of interest.

INPUT:

Input consists of two consecutive disc files in the cases of auto rotary bicoherence and cross rotary bicoherence of form XXY, and four such files in the case of cross rotary bicoherence of form XYY. (See reference 1 for explanation of notation.)

OUTPUT:

(1) Line Printer

Sample line printer output appears in Appendix 1. A facsimile of three input cards in printed, followed by a listing of the contour levels from Card 2 under a separate heading, "CONTOUR LEVELS."

(2) Plots

Consider the following region in the $\omega_1\omega_2$ plane:

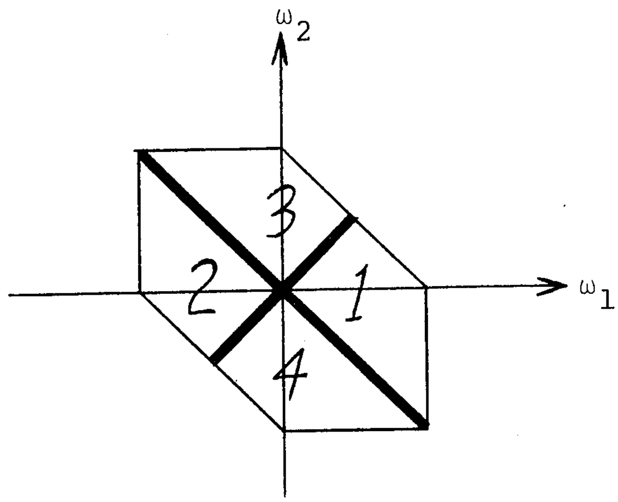


Figure 1

In terms of the notation of the report on program BIVEC (ref. 1), which calculates the rotary bicoherences, the user will recall that auto rotary bicoherences and cross rotary bicoherences of form XXY are calculated over the domains 1 and 2 of figure 1, while cross rotary bicoherences of form XYY require the addition of 3 and 4. As mentioned above, program RBPLOT exists in two versions, RBPLOT05 and RBPLOT06. RBPLOT05 generates separate plot files of each region of figure 1 as appropriate; thus it considers region 1+2, or 1+2+3+4. The plots generated constitute Appendix 2. They are separately annotated, but only the region 1 plot has the data origin, contour levels, and processing history. Program RBPLOT06, on the other hand, generates a single plot similar in appearance to figure 1 and displayed as Appendix 3. Again it consists of regions 1+2, or 1+2+3+4 as appropriate.

The program is capable of producing disc plot files to be used for immediate or deferred Versatec plots, or a plot tape to be used for Calcomp plotting. See the W.H.O.I. "Handbook for Computer Users" for details.

USAGE:

(1) Control cards

Let aaa = account number

uuu = user number

ffi = input file names

ppp = plot file name

Assume element file RBPLOTR (object of RBPLOTS, say) is in account aaa. Assume a Versatec plot is to be produced in this job. Then the control card portion of the input deck is:

```
      !JOB aaa,uuu
      !LIMIT (TIME,t), (CORE,c)
      !ASSIGN F:1, (FILE,ff1)
      !ASSIGN F:2, (FILE,ff2)
    * !ASSIGN F:3, (FILE,ff3)
    * !ASSIGN F:4, (FILE,ff4)
   ** !ASSIGN F:5, (FILE,SCRATCH),(OUTIN)
      !LOAD (EF,(RBPLOTR), (PLOTDFER,3)),(UNSAT,(3))
      !RUN
      !DATA
    ----------------->  (DATA CARDS)
    † !PLOTV
```

(2) Data cards

Program RBPLOT has NAMELIST input followed by three data cards. The NAMELIST input must be terminated by an * card whether or not any NAMELIST variables are being input.

The NAMELIST parameters and their default values are as follows. In general, "switches" are set to 0 for "off," 1 for "on."

---

*Necessary only if plotting cross rotary bicoherences of form XYY.

**Necessary only for RBPLOT06, cross rotary bicoherences of form XYY.

†Plot size is set to Versatec paper for RBPLOT05, and 30" Calcomp paper for RBPLOT06.

NAMELIST Variable	Meaning	Default Values
ISTORE1,ISTORE2 ISTORE3,ISTORE4	DCB assignments for input consecutive files	1,2,3,4 in order
ISCR	DCB assignment for scratch file	5
NDECPL	Format control for axis annotation: >0: Number of digits to the right of the decimal point which are plotted, after rounding =0: Only integer portion of number and decimal point are plotted, after rounding =-1: Only integer portion of number is plotted, after rounding <-1: |NDECPL|-1 digits are truncated from the integer portion after rounding	1
WIDTH	Width, in inches, of plot of L/2 points for RBPLOT05, L points for RBPLOT06, where L=piece length	8.75 for RBPLOT05 24.0 for RBPLOT06
CMPPT	Spacing in centimeters between adjacent data (grid) points on plot. If entered by user, this will override any specification of variable WIDTH	WIDTH*2.54/(L/2) (L=pc. length)
IBUOY	Switch to print information about data origin on plot, including original buoy format file names and variable numbers	1
IFOURIER	Switch to print processing history on plot (from program FOURIER)	1

The data cards follow the *card which terminates NAMELIST input. They are all in generalized, free-field format and are as follows:

Card 1

Variable No.	Meaning
1,2,3	This sequence of integers gives the form of auto or rotary cross bicoherence; e.g., rotary cross of form XYY is 122, and rotary auto is 111 or 222.
LPIECE	Piece length
SAMPSEC	Sample interval in seconds
FREQTIC	* Distance between tic marks on plot in $(hr)^{-1}$

Card 2

Variable No.	Meaning
1 (NCONLV)	Number of contour levels to follow (≤ 10)
2 thru (1+NCONLV)	NCONLV contour levels for plot
(2+NCONLV) thru (1+2·NCONLV)	NCONLV confidence levels (percent) for contour levels

Card 3

-up to 72 characters of identification, which will appear immediately above the plot. The first 36 characters will appear on one line, and the next 36 characters on the next line. The user is responsible for a proper transition of text from one line to the next, occurring between columns 36 and 37.

*Full-scale frequency (Nyquist frequency) is 1800/SAMPSEC. Variable FREQTIC does not have to divide evenly into the full-scale frequency.

RESTRICTIONS:

(1) Program RBPLOT uses a large two-dimensional buffer, "DATA," which can be thought of as a matrix whose elements correspond one-for-one with the grid points of the plot, and whose rectangular "boundary" completely contains the plot, the non-data points being padded with flags to that effect. The dimensions of DATA therefore govern the maximum piece length which can be plotted. The rule is as follows: If L is the piece length, then array DATA must be dimensioned at least DATA(R,C) where:

$$C \geq 1/2$$

$$R \geq 1.5C + L$$

The "standard" dimensions of DATA are DATA(200,130). If recompilation is necessary, the following three additional steps must be taken:

a) Array XDATA(C,R) is EQUIVALENCE'd to DATA, and must have dimensions exactly the reverse of those of DATA.

b) Variables MD1 and MD2 (immediately following the above EQUIVALENCE), must be initialized to the first and second dimensions of DATA respectively.

c) Check that array EXCH is DIMENSIONed the same as the first dimension of DATA.

(2) The number of contour levels must be ≤ 10.

SUBPROGRAMS REQUIRED: PLOTDFER, for a Versatec or graphics terminal plot.

STORAGE REQUIREMENT AND TIMING:

In a RBPLOT run in which PLOTDFER was also loaded, and array DATA was dimensioned DATA(200,130), a peak core of 77 512-word pages was used, along with 1.08 minutes of CPU time. This was to plot auto rotary bicoherence.

ERRORS AND DIAGNOSTICS:

None

PROGRAMMER:

Gerard H. Martineau

ORIGINATOR:

Melbourne G. Briscoe

DATE:

October, 1977

REFERENCES:

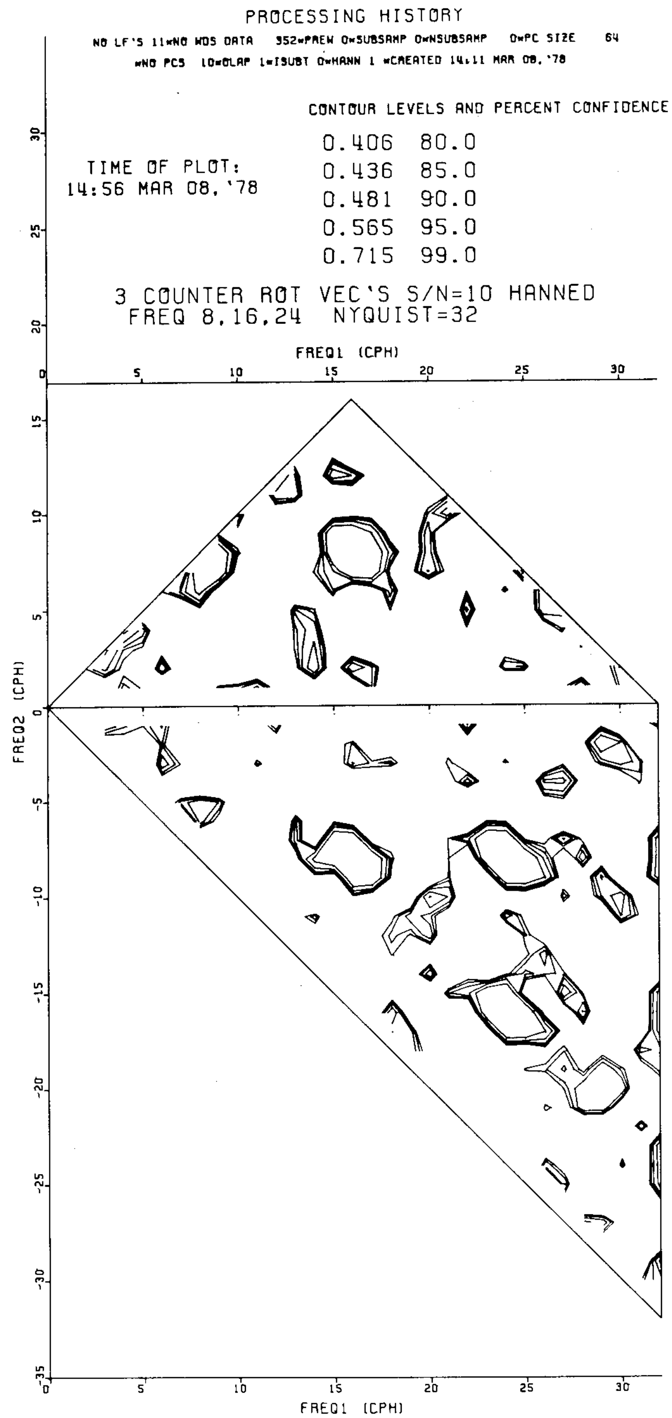
(1) Report on program BIVEC, by Gerard H. Martineau, Woods Hole Oceanographic Institution.

APPENDIX 1
(RBPLOT)

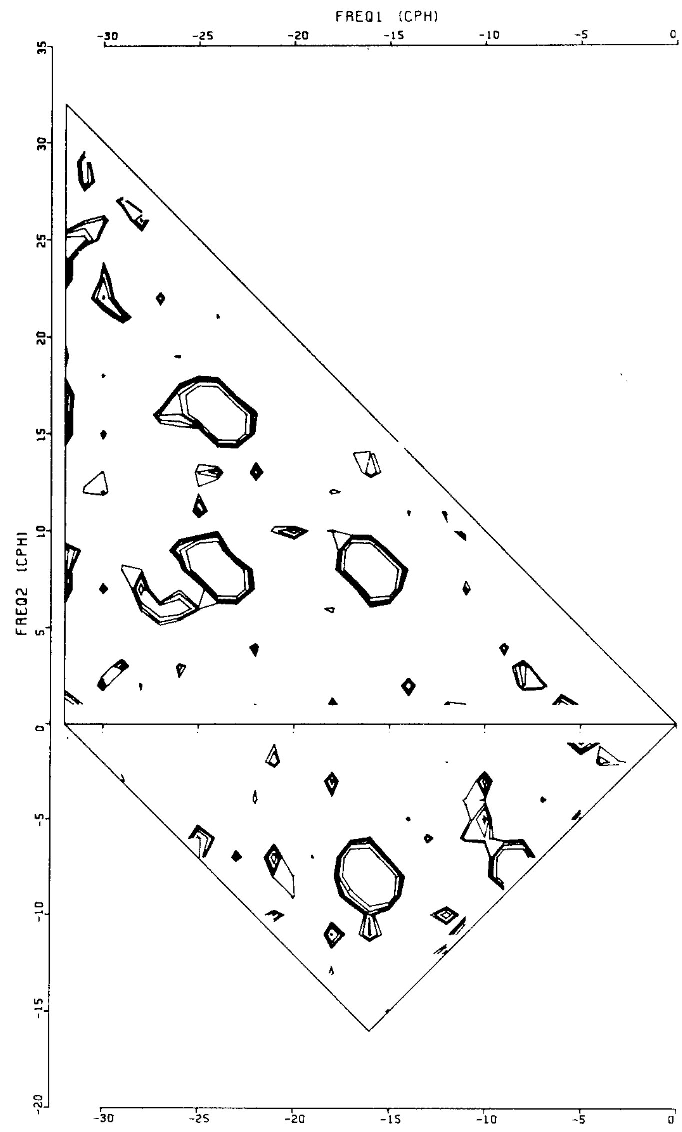
```
10:20 DEC 02,'77 ID=05B5
JØB 462,1719,7  , TERMINAL JØB
LIMIT (CØRE,4),(TIME,1),(LØ,50),(DØ,1)
PCL
C RBPLØTO5LP TØ LP
 1 1 1 64 56.25 5
 5 .405 .442 .485 .548 .705 80 85 90 95 99
 4 CØUNTER RØT VEC'S AMP SAME AS G N     FREQ 3/4,9/16,3/8,3/16 NYQUIST

CØNTØUR LEVELS:
       .405          .442          .485          .548          .705
```

APPENDIX 2, PART 1
(RBPLOT)

AUTO ROTARY BICOHERENCE



APPENDIX 2, PART 2
(RBPLOT)



APPENDIX 3
(RBPLOT)

AUTO ROTARY BICOHERENCE

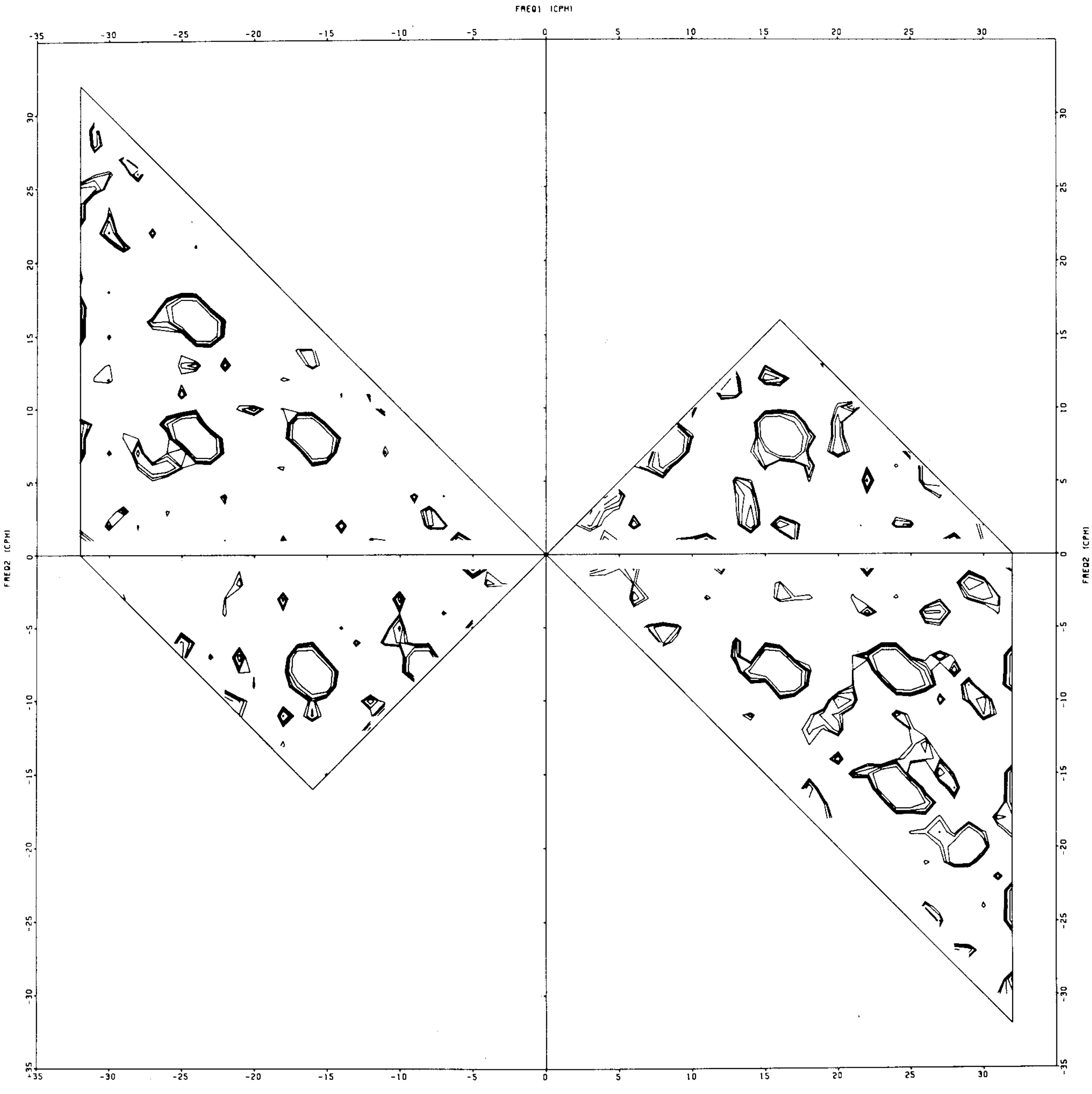
CONTOUR LEVELS AND PERCENT CONFIDENCE

0.406	80.000
0.436	85.000
0.481	90.000
0.565	95.000
0.715	99.000

3 COUNTER ROT VEC'S S/N=10 HANNED
FREQ 8,16,24 NYQUIST=32

PROCESSING HISTORY
NO LF'S 11*NO WDS DATA 352*PREW 0*
SUBSAMP 0*NSUBSAMP 0*PC SIZE 64*NO PCS 10*
OLAP 1*ISUBT 0*HANN 1 *CREATED 14:28 DEC 07,'77

TIME OF PLOT:
15:43 DEC 07,'77



E. FORTRAN IV LISTINGS, ALPHABETICALLY

- PROGRAM BISCAL -

```
C              <<<<<<  B I S C A L 0 4  >>>>>>
C CALCULATES BISPECTRA AND CROSS-BISPECTRA FOR ONE OR TWO
C REAL SERIES AND DETERMINES CONFIDENCE LEVELS.GENERATES DISC
C FILE FOR TRANSMITTAL TO PLOTTING PROGRAM.
C   PROGRAMMER:  G. MARTINEAU                         10/13/77
C
C
       DIMENSION OM1(5400),OM2(5400),SQ(70,70),
      $W3(300),KBIN(200),ABIN(200),CONF(10),BCL(10),KARD(20),
      $LABL(33),PHSERR(2)
       EQUIVALENCE (KBIN,ABIN)
       DOUBLE PRECISION R1,R2,R3,C1,C2,C3,S1,S2,S3,BR,BC,BMOD,
      $BRP,BCP
       NAMELIST IDATSTART,MF,ISTORE,
      $BINSIZE,BICOHLIM,IHI,KB1,KB2,KONF,LISTBI,IDUMP,INORM,
      $ISQOUT,IHANN,SIGLVL
       CALL ABORTSET(700S,1)
       PI=3.14159265;TWOPI=2.*PI
       IDUMP=1
       MF=21;IDATSTART=5
       ISTORE=1
       IHI=1
       M33=0
       BINSIZE=.01;BICOHLIM=1.5;KB1=KB2=1;KONF=0
       SIGLVL=-1 ;           PHSERR(2)=0.
       KOVER=0 ; LISTBI=0
       MTOT=0
       INORM=0
       ISQOUT=0 ; IHANN=1
C      OUTPUT 'ONE',MTOT
C
       DO 20 I=1,200
    20 KBIN(I)=0
C
       ISQD=70
       INPUT
       CALL DPAR(NERR,MF,1,1,KSUM,512)
       OUTPUT IDATSTART,MF,ISTORE,BINSIZE,
      $BICOHLIM,IHI,KB1,KB2,KONF,LISTBI,IDUMP,INORM,IHANN
C CHECK EXISTENCE OF DCB ASSIGNMENT FOR OUTPUT FILE
       IF (IDUMP.NE.0) CALL GETDCB(ISTORE,LOCC)
       READ (105,1010) KARD
  1010 FORMAT (20A4)
       WRITE (108,1020) KARD
  1020 FORMAT (1H ,20A4)
       DECODE (80,1050,KARD) LFW1,LFW2,LFW1P2,NPIECES,
      $LPIECE,NPC1,NPC2,CONF1,CONF2
  1050 FORMAT (9G)
       IF (KONF.EQ.0) GO TO 23
       READ (105,1010) KARD
       WRITE (108,1020) KARD
       DECODE (80,1200,KARD) NCONF,NCONF,(CONF(I),I=1,NCONF)
  1200 FORMAT (G,NG)
    23 CONTINUE
       IF (KONF.NE.0) NBINS=INT(BICOHLIM/BINSIZE+0.5);
      $BICOHLIM=NBINS*BINSIZE ;
      $WRITE (108,1600) BICOHLIM
  1600 FORMAT ('ADJUSTED BICOHERENCE LIMIT, HOLDING BINSIZE ',
```

```
        $'AS INPUT ',F6.2)
         IF ((LISTBI.NE.0).AND.(INORM.EQ.0)) WRITE (108,1321)
 1321 FORMAT (/,T3,'F1',T9,'F2',T15,'F3',T22,'BIC',T30,'BIPH',
        $T40,'BIMOD',T52,'BISPECR',T65,'BISPECI',T78,'AUTOF1',
        $T90,'AUTOF2',T101,'AUTOF1F2',T117,'SD',T125,'PHSERR',/)
         IF ((LISTBI.NE.0).AND.(INORM.NE.0)) WRITE (108,1431)
 1431 FORMAT (/,T3,'F1',T9,'F2',T15,'F3',T22,'BIC',T30,'BIPH',
        $T41,'BISPPR',T55,'BISPPI',T65,'PHSERR',/)
         NPCS=NPC2-NPC1+1
         IF ((NPCS.EQ.1).AND.(IHANN.EQ.1)) IHANN=0 ; OUTPUT
        $'IHANN HAS BEEN SET TO 0. CAN''T OVERLAP WITH ONE PIECE'
         IF (IHANN.EQ.1) EDOF=36.*NPCS*NPCS/(19*NPCS-1)
         IF (IHANN.EQ.0) EDOF=2*NPCS
         IF (SIGLVL.LT.0.) SIGLVL=SQRT(6./EDOF)
         LPHALF=LPIECE/2
         IF (MOD(LPHALF,2).EQ.1)
        $STOP '(PIECE LENGTH)/2 MUST BE EVEN'
         NFHALF=LPHALF/2
         IF (LISTBI.EQ.0) GO TO 25
   25 CONTINUE
C DECIDE WHETHER TO CALCULATE CROSS-BISPECTRA
        NSERIES=2
        IF ((LFW1.NE.LFW2).AND.(LFW1P2.NE.LFW1).AND.
       $(LFW1P2.NE.LFW2)) OUTPUT
       $'CANNOT HAVE MORE THAN TWO INPUT SERIES' ; STOP
        IF ((LFW1.EQ.LFW2).AND.(LFW2.EQ.LFW1P2)) NSERIES=1
        ISQ=1
C CHECK INPUT INFO. AGAINST LABELS
C       OUTPUT 'MK 1',IDATSTART
C
C
        NFREQLO=NFHALF+1
        NFREQHI=NFHALF
        NPTSLO=NPIECES*(2*NFREQLO-1)
        NPTSHI=NPIECES*(2*NFREQHI-1)
C
C
   30 CONTINUE
C
C SET ARRAY SQ TO -999.
C
        IF (IDUMP.EQ.0) GO TO 41
C
        DO 40 I=1,ISQD
C
        DO 40 J=1,ISQD
   40 SQ(I,J)=-999.
C
   41 CONTINUE
C
C       OUTPUT LFW1,IDATSTART,NPTSLO,NPTSHI,ISQ
        IF (ISQ.EQ.1) CALL RDISC(LFW1,IDATSTART,OM1,NPTSLO)
C       OUTPUT 'MK 3'
        IF ((ISQ.NE.3).OR.(NSERIES.NE.2).OR.(LFW1.NE.LFW2))
       $GO TO 46
        DO 43 I=1,NPTSLO
   43 OM2(I)=OM1(I)
   46 CONTINUE
        IF (ISQ.EQ.3)CALL RDISC(LFW1,IDATSTART+NPTSLO,OM1,NPTSHI)
C       OUTPUT 'MK 4'
```

```
      IF ((ISQ.NE.1).OR.(NSERIES.NE.1)) GO TO 75
C
C
      DO 50 I=1,NPTSLO
   50 OM2(I)=OM1(I)
C
C
      GO TO 100
   75 CONTINUE
      IF (NSERIES.EQ.1) GO TO 100
      IF ((ISQ.EQ.1).AND.(LFW1.NE.LFW2))
     $CALL RDISC(LFW2,IDATSTART,OM2,NPTSLO)
      IF ((ISQ.EQ.1).AND.(LFW1.EQ.LFW2))
     $CALL RDISC(LFW1P2,IDATSTART,OM2,NPTSLO)
      IF ((ISQ.EQ.5).AND.(LFW1.NE.LFW2))
     $CALL RDISC(LFW2,IDATSTART+NPTSLO,OM2,NPTSHI)
      IF ((ISQ.EQ.5).AND.(LFW1.EQ.LFW2))
     $CALL RDISC(LFW1P2,IDATSTART,OM2,NPTSLO)
C
  100 CONTINUE
C
C     OUTPUT 'MK 7'
      CALL RASTER
      NCPT=NFIRST
C     OUTPUT 'MK 8',NCPT
  105 CONTINUE
C     OUTPUT 'MK 9'
      CALL FREQINIT
  110 CONTINUE
C     OUTPUT 'MK 10'
      IF3= IF1+IF2
      IF2A=IABS(IF2)
      IF3A=IABS(IF3)
      IST3=IDATSTART+(2*(IF3A-1)+1)*NPIECES
      NR3=2*NPIECES
      IF (IF3A.EQ.LPHALF) NR3=NPIECES
      IF (IF3A.EQ.0) NR3=NPIECES
      IF (IF3A.GT.LPHALF) STOP 'CHKPT 1; CHK LOGIC'
C READ IN SUM FREQUENCY COEFFICIENTS IF NECESSARY
C     OUTPUT 'MK 101',ISQ
      CALL SUMFREQ
C     OUTPUT 'MK 102',IREAD
      IF (IREAD.EQ.0) GO TO 120
C DON'T READ SUM FREQ IF ALREADY HAVE IN THIS SCAN
      IF (M33.EQ.1) GO TO 160
C     OUTPUT 'MK 11'
C     OUTPUT LFW1P2,IST3,NR3
      CALL RDISC(LFW1P2,IST3,W3,NR3)
      M33=1
      GO TO 160
C 117 OUTPUT 'MK 13'
C
  120 CONTINUE
C     OUTPUT 'MK 131'
C
C SUM FREQ COEFF'S WILL COME FROM OM1 OR OM2
      IF (ISQ.GT.2) GO TO 138
      IST3=(2*IF3A-1)*NPIECES+1
      IF (NSERIES.NE.1) GO TO 128
      DO 125 I=1,NR3
```

```
  125 W3(I)=OM1(IST3+I-1)
      GO TO 160
  128 CONTINUE
      IF (LFW1.EQ.LFW2) GO TO 133
      IF ((LFW1.NE.LFW2).AND.(LFW1P2.EQ.LFW2)) GO TO 133
      DO 130 I=1,NR3
  130 W3(I)=OM1(IST3+I-1)
      GO TO 160
  133 CONTINUE
      DO 135 I=1,NR3
  135 W3(I)=OM2(IST3+I-1)
      GO TO 160
  138 IF (ISQ.GT.3) GO TO 148
      IST3=2*(IF3A-NFHALF-1)*NPIECES+1
      IF (NSERIES.NE.1) GO TO 143
      DO 140 I=1,NR3
  140 W3(I)=OM1(IST3+I-1)
      GO TO 160
  143 CONTINUE
      IF (LFW1P2.EQ.LFW2) GO TO 148
      DO 145 I=1,NR3
  145 W3(I)=OM1(IST3+I-1)
      GO TO 160
  148 CONTINUE
      IF (ISQ.EQ.5) GO TO 155
      IF (LFW1P2.EQ.LFW2) GO TO 152
      IST3=2*(IF3A-NFHALF-1)*NPIECES+1
      DO 150 I=1,NR3
  150 W3(I)=OM1(IST3+I-1)
      GO TO 160
  152 IST3=(2*IF3A-1)*NPIECES+1
      DO 154 I=1,NR3
  154 W3(I)=OM2(IST3+I-1)
      GO TO 160
  155 CONTINUE
      IST3=(2*IF3A-1)*NPIECES+1
      IF (LFW1.NE.LFW2) GO TO 160
      DO 157 I=1,NR3
  157 W3(I)=OM2(IST3+I-1)
      GO TO 160
C
C
  160 CONTINUE
C     OUTPUT 'MK 16'
C
      IST1=1+(2*(IF1-1)+1)*NPIECES
      IF (ISQ.GT.2) IST1=2*(IF1-NFHALF-1)*NPIECES+1
      IST2=1+(2*(IF2A-1)+1)*NPIECES
      IF (ISQ.EQ.5) IST2=2*(IF2A-NFHALF-1)*NPIECES+1
      IF (IF1.EQ.0) IST1=1
      IF (IF2A.EQ.0) IST2=1
C CALCULATE BISPECTRAL COEFFICIENTS
      BR=BC=S1=S2=S3=0.
      ONE=1.
      IF (IF2.LT.0) ONE=-1.
C
C
C     OUTPUT 'MK 17',IST1,IST2
      DO 200 I=NPC1,NPC2
      R1=OM1(IST1+I-1)
```

```
      C1=OM1(IST1+NPIECES+I-1)
      IF ((IF1.EQ.0).OR.(IF1.EQ.LPHALF))  C1=0.
      IF ((LFW1.EQ.LFW2).AND.(NSERIES.EQ.2).AND.
     $(ISQ.NE.3).AND.(ISQ.NE.4)) GO TO 180
      R2=OM2(IST2+I-1)
      C2=OM2(IST2+NPIECES+I-1)
      GO TO 190
  180 CONTINUE
      R2=OM1(IST2+I-1) ; C2=OM1(IST2+NPIECES+I-1)
  190 CONTINUE
      IF ((IF2A.EQ.0).OR.(IF2A.EQ.LPHALF))  C2=0.
      R3=W3(I)
      C3=W3(I+NPIECES)
      IF ((IF3A.EQ.0).OR.(IF3A.EQ.LPHALF))  C3=0.
      IF (INORM.EQ.0) GO TO 195
      BRP=      R1*R2*R3+ONE*R1*C2*C3+C1*R2*C3-ONE*C1*C2*R3
      BCP=  ONE*R1*C2*R3-     R1*R2*C3+C1*R2*R3+ONE*C1*C2*C3
      BMOD=DSQRT(BRP**2+BCP**2)
      BR=BR+BRP/BMOD ; BC=BC+BCP/BMOD
      GO TO 200
  195 CONTINUE
      BR=BR+     R1*R2*R3+ONE*R1*C2*C3+C1*R2*C3-ONE*C1*C2*R3
      BC=BC+ONE*R1*C2*R3-     R1*R2*C3+C1*R2*R3+ONE*C1*C2*C3
      S1=S1+R1*R1+C1*C1
      S2=S2+R2*R2+C2*C2
      S3=S3+R3*R3+C3*C3
  200 CONTINUE
C
C
      IF (INORM.EQ.0) GO TO 202
      BICOH=(DSQRT(BR**2+BC**2))/NPCS
      GO TO 204
  202 CONTINUE
      BR=BR/NPCS ; BC=BC/NPCS
      S1=S1/NPCS ; S2=S2/NPCS ; S3=S3/NPCS
C     OUTPUT S1,S2,S3,NPCS
      SD=DSQRT(S1*S2*S3)
C     OUTPUT BR,BC,SD
      BICOHR=BR/SD
      BICOHC=BC/SD
      BICOH=SQRT(BICOHR*BICOHR+BICOHC*BICOHC)
  204 CONTINUE
      BIPHSE=DATAN2(BC,BR)
      BIPHSE=BIPHSE*360./TWOPI
C     OUTPUT 'TWO',MTOT
      IF (KONF.NE.0) CALL CBIN
C     OUTPUT BICOH
      BIMOD=DSQRT(BR*BR+BC*BC)
      IF (LISTBI.EQ.0) GO TO 215
      IF (INORM.NE.0) GO TO 212
      IF ((BICOH.LE.CONF1).OR.(BICOH.GT.CONF2)) GO TO 215
      IF (IHANN.NE.1) GO TO 205
      IF (BICOH.GE.SIGLVL) NPH=1 ;
     $PHSERR(1)=57.296*ASIN(1.96/SQRT(EDOF)/BICOH) ; GO TO 206
      NPH=0 ; GO TO 206
  205 CONTINUE
      IF ((IHANN.EQ.0).AND.(BICOH.GE.SIGLVL)) NPH=1 ;
     $PHSERR(1)=SQRT(2.*NPCS) ; GO TO 206
      NPH=0
  206 CONTINUE
```

```
      WRITE (108,2000) IF1,IF2,IF3,BICOH,BIPHSE,BIMOD,BR,BC,
     $S1,S2,S3,SD,NPH,(PHSERR(I),I=1,NPH)
 2000 FORMAT (X,I4,2(2X,I4),3X,F5.3,3X,F6.0,-X,1H ,3(2X,E11.3),
     $3(1X,E11.3),2X,E11.3,3X,N(F5.1))
      GO TO 215
  212 CONTINUE
      IF ((BICOH.LE.CONF1).OR.(BICOH.GT.CONF2)) GO TO 215
      IF (IHANN.NE.1) GO TO 213
      IF (BICOH.GE.SIGLVL) NPH=1 ;
     $PHSERR(1)=57.296*ASIN(1.96/SQRT(EDOF)/BICOH) ; GO TO 214
      NPH=0 ; GO TO 214
  213 CONTINUE
      IF ((IHANN.EQ.0).AND.(BICOH.GE.SIGLVL)) NPH=1 ;
     $PHSERR(1)=SQRT(2.*NPCS) ; GO TO 214
      NPH=0
  214 CONTINUE
      WRITE (108,2100) IF1,IF2,IF3,BICOH,BIPHSE,
     $BR/NPIECES,BC/NPIECES,NPH,(PHSERR(I),I=1,NPH)
 2100 FORMAT (X,I4,2(2X,I4),3X,F5.3,3X,F6.0,-X,1H ,
     $2(3X,E11.3),3X,N(F5.1))
  215 CONTINUE
C     OUTPUT IF1
C COMMIT BICOH TO STORAGE BUFFER SQ
      IF (IDUMP.EQ.0) GO TO 242
      CALL INDEX
      SQ(NROW,NCOL)=BICOH
  242 CONTINUE
C INCREMENT FREQUENCIES
      IF1=IF1+1     ; IF2=IF2-1
C END OF SCAN?
      CALL BOUNDARY
      GO TO (110,250),IBDY+1
C INCREMENT SCAN
  250 NCPT=NCPT+1 ; M33=0
C     WRITE (108,2222) IF1,ISQ,NCPT
C2222 FORMAT (1H ,'IF1=',I3,2X,'ISQ=',I2,2X,'NCPT=',I3)
C END OF RASTER?
      IF (NCPT.LE.NLAST) GO TO 105
C OUTPUT TO DISC
      IF (IDUMP.EQ.0) GO TO 260
      WRITE (ISTORE) ((SQ(I,J),I=1,NFHALF),J=1,NFHALF)
  260 CONTINUE
C INCREMENT INDEX FOR AREA BEING CONSIDERED (ISQ)
      IF (ISQOUT.NE.0) OUTPUT ISQ
      ISQ=ISQ+1
      IF ((NSERIES.EQ.1).AND.(ISQ.EQ.2)) ISQ=ISQ+1 ; GO TO 30
      IF ((NSERIES.EQ.1).AND.(ISQ.EQ.4)) GO TO 400
      IF (ISQ.LE.5) GO TO 30
C
  400 CONTINUE
C TRANSFER FOURIER LABELS TO BICOHERENCE FILE IF APPROPRIATE
      IF (IDUMP.EQ.0) GO TO 450
      IF (IDATSTART.NE.5) GO TO 420
      CALL RDISC(LFW1,1,LABL,4)
      WRITE (ISTORE) (LABL(I),I=1,4)
      CALL RDISC(LFW2,1,LABL,4)
      WRITE (ISTORE) (LABL(I),I=1,4)
      CALL RDISC(LFW1P2,1,LABL,4)
      WRITE (ISTORE) (LABL(I),I=1,4)
      WRITE (ISTORE) MF
```

```
      GO TO 440
  420 CONTINUE
      DO 425 I=1,3
  425 LABL(I)=4H
      LABL(4)=0
      DO 430 KZ=1,3
  430 WRITE (ISTORE) (LABL(I),I=1,4)
      WRITE (ISTORE) MF
  440 CONTINUE
      IF (MF.NE.21) GO TO 450
      CALL RDISC(21,1,LABL,33)
      WRITE (ISTORE) LABL
  450 CONTINUE
C     OUTPUT 'THREE',MTOT
      IF (KONF.NE.0) CALL CONFIDENCE
C
C
      GO TO 750
  700 STOP 'ABORTSET TERMINATION'
  750 STOP 'NORMAL PROGRAM COMPLETION'
C
C
      SUBROUTINE BOUNDARY
C INTERNAL SUBROUTINE FOR PROGRAM BISCAL03 TO TEST
C SCAN FOR FREQUENCY LIMITS
      IBDY=1
      GO TO (100,200,300,400,500),ISQ
  100 IF((IF1.LE.NFHALF).AND.(IF2.GT.0)) IBDY=0
      RETURN
  200 IF (IF1.LE.NFHALF) IBDY=0
      RETURN
  300 IF (IF2.GT.0) IBDY=0
      RETURN
  400 IF ((NCPT.LE.NFHALF).AND.(IF2.GE.-NFHALF)) IBDY=0
      IF ((NCPT.GT.NFHALF).AND.(IF1.LE.LPHALF)) IBDY=0
      RETURN
  500 IF (IF1.LE.LPHALF) IBDY=0
      RETURN
C
C
      SUBROUTINE FREQINIT
C INTERNAL SUBROUTINE FOR PROGRAM BISCAL03 TO INITIALIZE
C FREQUENCIES FOR A GIVEN LINE OF SCAN
      GO TO (100,200,300,400,500),ISQ
  100 IF1=NCPT/2 ; IF2=NCPT/2
      IF (MOD(NCPT,2).EQ.1) IF1=IF1+1
      RETURN
  200 IF1=NCPT+1 ; IF2=-1
      RETURN
  300 IF1=NFIRST-1 ; IF2=NCPT-IF1
      RETURN
  400 CONTINUE
      IF (NCPT.LE.NFHALF) IF1=NFHALF+1 ;
     $IF2=-IF1+NCPT
      IF (NCPT.GT.NFHALF) IF1=NCPT+1 ; IF2=-1
      RETURN
  500 IF1=NCPT+NFHALF+1 ; IF2=-NFHALF-1
      RETURN
C
C
```

```
      SUBROUTINE INDEX
C INTERNAL SUBROUTINE FOR PROGRAM BISCAL03 TO RELATE
C FREQUENCIES TO STORAGE ARRAY INDICES
      GO TO (100,200,300,400,500),ISQ
  100 NROW=IF2 ; NCOL=IF1
      RETURN
  200 NROW=IF2+NFHALF+1 ; NCOL=IF1
      RETURN
  300 NROW=IF2 ; NCOL=IF1-NFHALF
      RETURN
  400 NROW=IF2+NFHALF+1 ; NCOL=IF1-NFHALF
      RETURN
  500 NROW=IF2+2*NFHALF+1 ; NCOL=IF1-NFHALF
      RETURN
C
C
      SUBROUTINE RASTER
C INTERNAL SUBROUTINE FOR PROGRAM BISCAL03 TO SET UP SCAN
C PARAMETERS
      GO TO (100,200,300,400,500),ISQ
  100 NFIRST=2 ; NLAST=LPHALF
      RETURN
  200 NFIRST=1 ; NLAST=NFHALF-1
      RETURN
  300 NFIRST=NFHALF+2 ; NLAST=LPHALF
      RETURN
  400 NFIRST=1 ; NLAST=LPHALF-1
      RETURN
  500 NFIRST=1 ; NLAST=NFHALF-1
      RETURN
C
C
      SUBROUTINE SUMFREQ
C INTERNAL SUBROUTINE FOR PROGRAM BISCAL03 TO DETERMINE WHERE
C TO FIND SUM FREQUENCY COEFFICIENTS
C RETURNS IREAD=1 IF READING IS NECESSARY, 0 OTHERWISE
      IREAD=0
      GO TO (100,400,150,200,300),ISQ
  100 IF (IF3A.GT.NFHALF) IREAD=1
      RETURN
  150 CONTINUE
      IF (NSERIES.EQ.1) RETURN
      IF (LFW1.EQ.LFW2) IREAD=1 ; RETURN
      IF ((LFW1.NE.LFW2).AND.(LFW1P2.EQ.LFW2)) IREAD=1 ; RETURN
      RETURN
  200 CONTINUE
      IF ((LFW1P2.EQ.LFW1).AND.(IF3A.LE.NFHALF).AND.
     $(LFW1.NE.LFW2)) IREAD=1 ; RETURN
      IF ((LFW1P2.EQ.LFW2).AND.(IF3A.GT.NFHALF).AND.
     $(LFW1.NE.LFW2)) IREAD=1 ; RETURN
      IF ((LFW1.EQ.LFW2).AND.(NSERIES.EQ.2)) IREAD=1 ; RETURN
      RETURN
  300 IF (LFW1.NE.LFW2) IREAD=1
  400 CONTINUE
      RETURN
C
C
      SUBROUTINE CBIN
C INTERNAL SUBROUTINE FOR PROGRAM BISCAL03 TO DETERMINE
C CONFIDENCE LIMITS FOR BICOHERENCES
```

```
C HAS ENTRY POINTS CBIN TO PLACE GIVEN BICOHERENCE IN
C BIN AND CONFIDENCE TO INITIATE CONFIDENCE LIMIT CALCULATION
C       OUTPUT 'FOUR',MTOT
        IF (BICOH.LE.BICOHLIM) GO TO 100
        KOVER=KOVER+1
        WRITE (108,1000) IF1,IF2,IF3,BICOH
 1000   FORMAT ('BICOH LIM EXCEEDED AT FREQ TRIPLET ',3(I3,X),
      $'  WITH VALUE ',G11.5)
        IF (IHI.NE.0) MTOT=MTOT+1
C       OUTPUT 'FIVE',MTOT
        RETURN
C
  100   CONTINUE
        NIB=INT(BICOH/BINSIZE)+1
        IF (NIB.GT.NBINS) GO TO 250
        KBIN(NIB)=KBIN(NIB)+1
        MTOT=MTOT+1
        RETURN
C
  250   CONTINUE
        WRITE (108,1025) BICOH,IF1,IF2,IF3
 1025   FORMAT ('BICOH OF ',F6.2,' AT FREQ TRIPLET ',3(I3,X),
      $' MET NEITHER BIN NOR LIM CRITERIA')
        MTOT=MTOT+1
        RETURN
C
C*****************************************************************
        ENTRY CONFIDENCE
C*****************************************************************
C OPTIONAL DISPLAY OF PARTITIONING OF BICOHERENCES
        IF (KB1.EQ.0) GO TO 320
        WRITE (108,1050)
 1050   FORMAT (/,T2,'BIN NO',T14,'BIN LIMITS',T32,'NO OF BIC''S'
      $,/)
C
C
        DO 300 I=1,NBINS
        WRITE (108,1100) I,(I-1)*BINSIZE,I*BINSIZE,KBIN(I)
 1100    FORMAT (T3,I3,T13,2(F4.2,4X),T34,I5)
  300   CONTINUE
C
C
  320   CONTINUE
C CONVERT NO OF BIC'S IN KBIN TO TOTAL FRACTION AT INDEX VALUE
        KSUBTOT=0 ; I=0
        WRITE (108,1125) MTOT
 1125   FORMAT (/,'TOTAL NO OF BIC''S SUBMITTED: ',I5,/)
        FTOT=FLOAT(MTOT)
C
  340   CONTINUE
        I=I+1
        IF (I.GT.NBINS) GO TO 350
C
        KSUBTOT=KSUBTOT+KBIN(I)
C       OUTPUT I,KSUBTOT,KBIN(I),FTOT
        ABIN(I)=FLOAT(KSUBTOT)
        ABIN(I)=ABIN(I)/FTOT
C       OUTPUT ABIN(I)
        GO TO 340
C
```

```
  350 CONTINUE
C OPTIONAL DISPLAY OF PARTIAL SUMS AT INDEX VALUE
      IF (KB2.EQ.0) GO TO 400
      WRITE (108,1150)
 1150 FORMAT (/,T32,'FRACT BELOW',/,T2,'BIN NO',T14,
     $'BIN LIMITS',T31,'UPPER BIN LIM',/)
C
C
      DO 380 I=1,NBINS
      WRITE (108,1200) I,(I-1)*BINSIZE,I*BINSIZE,ABIN(I)
 1200 FORMAT (T3,I3,T13,2(F4.2,4X),T34,F5.3)
  380 CONTINUE
C
C
  400 CONTINUE
C
C
C COMPUTE CONFIDENCE LIMITS
      I=0
C
  420 CONTINUE
      I=I+1
      IF (I.GT.NCONF) GO TO 600
      J=0
C
  450 CONTINUE
      J=J+1
      IF (J.GT.NBINS) WRITE (108,1250) CONF(I) ; BCL(I)=999. ;
     $GO TO 420
 1250 FORMAT ('BINS EXHAUSTED FOR CONFIDENCE LEVEL',F6.3)
      IF (ABIN(J).GE.CONF(I))  GO TO 470
      GO TO 450
C
  470 CONTINUE
C
      JH=J
      JL=J-1
      Y1=JL*BINSIZE
      Y2=JH*BINSIZE
      X1=0
      IF (JL.NE.0) X1=ABIN(JL)
      X2=ABIN(JH)
      X0=CONF(I)
      BCL(I)=Y1+(X0-X1)*(Y2-Y1)/(X2-X1)
      GO TO 420
  600 CONTINUE
C
C PRINT CONFIDENCE LEVELS
      WRITE (108,1400) NCONF,(CONF(I)*100.,BCL(I),I=1,NCONF)
 1400 FORMAT (/,T4,'CONFIDENCE',/,T6,'LEVEL',T21,'BICOHERENCE',
     $/,T5,'(PERCENT)',/,N(T7,F4.1,T24,F5.3,/),/)
      WRITE (108,1450) BICOHLIM,KOVER
 1450 FORMAT ('NO. OF BICOHERENCES GREATER THAN SPECIFIED ',
     $'MAXIMUM OF ',F5.2,' IS ',I5,/)
      RETURN
      END
```

- PROGRAM BISUM -

```
C            <<<<<<<  B I S U M 0 2  >>>>>>>
C CALCULATES BISPECTRA AND CROSS-BISPECTRA FOR ONE OR TWO
C REAL SERIES AND INTEGRATES THEM OVER CONSTANT SUM FREQUENCIES
C AND PRINTS FORMATTED SUMMARY
C   PROGRAMMER: GERARD H. MARTINEAU            1/19/78
C   ORIGINATOR:  MELBOURNE G. BRISCOE
      DIMENSION OM1(5400),OM2(5400),SQ(70,70),
     $W3(300),CONF(10),BCL(10),KARD(20),
     $LABL(33),PHSERR(2)
      DIMENSION BRU(130),BIU(130),BICU(130),BIC2U(130),
     $BCSU(130),BRL(130),BIL(130),BICL(130),BIC2L(130),
     $BCSL(130)
      DOUBLE PRECISION R1,R2,R3,C1,C2,C3,S1,S2,S3,BR,BC,BMOD,
     $BRP,BCP,BM
      NAMELIST IDATSTART,MF,ISTORE,
     $BINSIZE,BICOHLIM,IHI,KB1,KB2,LISTBI,IDUMP,INORM,
     $ISQOUT,IHANN,SIGLVL,KDCB,KDISC
      CALL ABORTSET(700S,1)
      PI=3.14159265;TWOPI=2.*PI
      IDUMP=0
      KDCB=2 ; KDISC=0
      MF=21;IDATSTART=5
      ISTORE=0
      JDIM=130
      IHI=1
      M33=0
      BINSIZE=.01;BICOHLIM=1.5;KB1=KB2=1;KONF=0
      SIGLVL=-1 ;           PHSERR(2)=0.
      KOVER=0 ; LISTBI=0
      MTOT=0
      INORM=0
      ISQOUT=0 ; IHANN=1
C     OUTPUT 'ONE',MTOT
C
C
C INITIALIZE OUTPUT ARRAYS
      DO 10 I=1,JDIM
      BRU(I)=BIU(I)=BIC2U(I)=BICU(I)=BCSU(I)=0.
      BRL(I)=BIL(I)=BIC2L(I)=BICL(I)=BCSL(I)=0.
   10 CONTINUE
C
C
      ISQD=70
      INPUT
      CALL DPAR(NERR,MF,1,1,KSUM,512)
      OUTPUT IDATSTART,MF,ISTORE,BINSIZE,
     $BICOHLIM,IHI,KB1,KB2,LISTBI,IDUMP,INORM,IHANN,
     $KDISC,KDCB
C CHECK EXISTENCE OF DCB ASSIGNMENT FOR OUTPUT FILE
      IF (IDUMP.NE.0) CALL GETDCB(ISTORE,LOCC)
      READ (105,1010) KARD
 1010 FORMAT (20A4)
      WRITE (108,1020) KARD
 1020 FORMAT (1H ,20A4)
      DECODE (80,1050,KARD) LFW1,LFW2,LFW1P2,NPIECES,
     $LPIECE,NPC1,NPC2,CONF1,CONF2
 1050 FORMAT (9G)
   23 CONTINUE
```

```
      IF ((LISTBI.NE.0).AND.(INORM.EQ.0)) WRITE (108,1321)
 1321 FORMAT (/,T3,'F1',T9,'F2',T15,'F3',T22,'BIC',T30,'BIPH',
     $T40,'BIMOD',T52,'BISPECR',T65,'BISPECI',T78,'AUTOF1',
     $T90,'AUTOF2',T101,'AUTOF1F2',T117,'SD',T125,'PHSERR',/)
      IF ((LISTBI.NE.0).AND.(INORM.NE.0)) WRITE (108,1431)
 1431 FORMAT (/,T3,'F1',T9,'F2',T15,'F3',T22,'BIC',T30,'BIPH',
     $T41,'BISPPR',T55,'BISPPI',T65,'PHSERR',/)
      NPCS=NPC2-NPC1+1
      IF ((NPCS.EQ.1).AND.(IHANN.EQ.1)) IHANN=0 ; OUTPUT
     $'IHANN HAS BEEN SET TO 0. CAN''T OVERLAP WITH ONE PIECE'
      IF (IHANN.EQ.1) EDOF=36.*NPCS*NPCS/(19*NPCS-1)
      IF (IHANN.EQ.0) EDOF=2*NPCS
      IF (SIGLVL.LT.0.) SIGLVL=SQRT(6./EDOF)
      LPHALF=LPIECE/2
      IF (MOD(LPHALF,2).EQ.1)
     $STOP '(PIECE LENGTH)/2 MUST BE EVEN'
      NFHALF=LPHALF/2
      IF (LISTBI.EQ.0) GO TO 25
   25 CONTINUE
C DECIDE WHETHER TO CALCULATE CROSS-BISPECTRA
      NSERIES=2
      IF ((LFW1.NE.LFW2).AND.(LFW1P2.NE.LFW1).AND.
     $(LFW1P2.NE.LFW2)) OUTPUT
     $'CANNOT HAVE MORE THAN TWO INPUT SERIES' ; STOP
      IF ((LFW1.EQ.LFW2).AND.(LFW2.EQ.LFW1P2)) NSERIES=1
      ISQ=1
C CHECK INPUT INFO. AGAINST LABELS
C     OUTPUT 'MK 1',IDATSTART
C
C
      NFREQLO=NFHALF+1
      NFREQHI=NFHALF
      NPTSLO=NPIECES*(2*NFREQLO-1)
      NPTSHI=NPIECES*(2*NFREQHI-1)
C
C
   30 CONTINUE
C
C SET ARRAY SQ TO -999.
C
      IF (IDUMP.EQ.0) GO TO 41
C
      DO 40 I=1,ISQD
C
      DO 40 J=1,ISQD
   40 SQ(I,J)=-999.
C
   41 CONTINUE
C
C     OUTPUT LFW1,IDATSTART,NPTSLO,NPTSHI,ISQ
      IF (ISQ.EQ.1) CALL RDISC(LFW1,IDATSTART,OM1,NPTSLO)
C     OUTPUT 'MK 3'
      IF ((ISQ.NE.3).OR.(NSERIES.NE.2).OR.(LFW1.NE.LFW2))
     $GO TO 46
      DO 43 I=1,NPTSLO
   43 OM2(I)=OM1(I)
   46 CONTINUE
      IF (ISQ.EQ.3)CALL RDISC(LFW1,IDATSTART+NPTSLO,OM1,NPTSHI)
C     OUTPUT 'MK 4'
      IF ((ISQ.NE.1).OR.(NSERIES.NE.1)) GO TO 75
```

```
C
C
      DO 50 I=1,NPTSLO
   50 OM2(I)=OM1(I)
C
C
      GO TO 100
   75 CONTINUE
      IF (NSERIES.EQ.1) GO TO 100
      IF ((ISQ.EQ.1).AND.(LFW1.NE.LFW2))
     $CALL RDISC(LFW2,IDATSTART,OM2,NPTSLO)
      IF ((ISQ.EQ.1).AND.(LFW1.EQ.LFW2))
     $CALL RDISC(LFW1P2,IDATSTART,OM2,NPTSLO)
      IF ((ISQ.EQ.5).AND.(LFW1.NE.LFW2))
     $CALL RDISC(LFW2,IDATSTART+NPTSLO,OM2,NPTSHI)
      IF ((ISQ.EQ.5).AND.(LFW1.EQ.LFW2))
     $CALL RDISC(LFW1P2,IDATSTART,OM2,NPTSLO)
C
  100 CONTINUE
C
C     OUTPUT 'MK 7'
      CALL RASTER
      NCPT=NFIRST
C     OUTPUT 'MK 8',NCPT
  105 CONTINUE
C     OUTPUT 'MK 9'
      CALL FREQINIT
  110 CONTINUE
C     OUTPUT 'MK 10'
      IF3= IF1+IF2
      IF2A=IABS(IF2)
      IF3A=IABS(IF3)
      IST3=IDATSTART+(2*(IF3A-1)+1)*NPIECES
      NR3=2*NPIECES
      IF (IF3A.EQ.LPHALF) NR3=NPIECES
      IF (IF3A.EQ.0) NR3=NPIECES
      IF (IF3A.GT.LPHALF) STOP 'CHKPT 1; CHK LOGIC'
C READ IN SUM FREQUENCY COEFFICIENTS IF NECESSARY
C     OUTPUT 'MK 101',ISQ
      CALL SUMFREQ
C     OUTPUT 'MK 102',IREAD
      IF (IREAD.EQ.0) GO TO 120
C DON'T READ SUM FREQ IF ALREADY HAVE IN THIS SCAN
      IF (M33.EQ.1) GO TO 160
C     OUTPUT 'MK 11'
C     OUTPUT LFW1P2,IST3,NR3
      CALL RDISC(LFW1P2,IST3,W3,NR3)
      M33=1
      GO TO 160
C 117 OUTPUT 'MK 13'
C
  120 CONTINUE
C     OUTPUT 'MK 131'
C
C SUM FREQ COEFF'S WILL COME FROM OM1 OR OM2
      IF (ISQ.GT.2) GO TO 138
      IST3=(2*IF3A-1)*NPIECES+1
      IF (NSERIES.NE.1) GO TO 128
      DO 125 I=1,NR3
  125 W3(I)=OM1(IST3+I-1)
```

```
      GO TO 160
  128 CONTINUE
      IF (LFW1.EQ.LFW2) GO TO 133
      IF ((LFW1.NE.LFW2).AND.(LFW1P2.EQ.LFW2)) GO TO 133
      DO 130 I=1,NR3
  130 W3(I)=OM1(IST3+I-1)
      GO TO 160
  133 CONTINUE
      DO 135 I=1,NR3
  135 W3(I)=OM2(IST3+I-1)
      GO TO 160
  138 IF (ISQ.GT.3) GO TO 148
      IST3=2*(IF3A-NFHALF-1)*NPIECES+1
      IF (NSERIES.NE.1) GO TO 143
      DO 140 I=1,NR3
  140 W3(I)=OM1(IST3+I-1)
      GO TO 160
  143 CONTINUE
      IF (LFW1P2.EQ.LFW2) GO TO 148
      DO 145 I=1,NR3
  145 W3(I)=OM1(IST3+I-1)
      GO TO 160
  148 CONTINUE
      IF (ISQ.EQ.5) GO TO 155
      IF (LFW1P2.EQ.LFW2) GO TO 152
      IST3=2*(IF3A-NFHALF-1)*NPIECES+1
      DO 150 I=1,NR3
  150 W3(I)=OM1(IST3+I-1)
      GO TO 160
  152 IST3=(2*IF3A-1)*NPIECES+1
      DO 154 I=1,NR3
  154 W3(I)=OM2(IST3+I-1)
      GO TO 160
  155 CONTINUE
      IST3=(2*IF3A-1)*NPIECES+1
      IF (LFW1.NE.LFW2) GO TO 160
      DO 157 I=1,NR3
  157 W3(I)=OM2(IST3+I-1)
      GO TO 160
C
C
  160 CONTINUE
C     OUTPUT 'MK 16'
C
      IST1=1+(2*(IF1-1)+1)*NPIECES
      IF (ISQ.GT.2) IST1=2*(IF1-NFHALF-1)*NPIECES+1
      IST2=1+(2*(IF2A-1)+1)*NPIECES
      IF (ISQ.EQ.5) IST2=2*(IF2A-NFHALF-1)*NPIECES+1
      IF (IF1.EQ.0) IST1=1
      IF (IF2A.EQ.0) IST2=1
C CALCULATE BISPECTRAL COEFFICIENTS
      BR=BC=S1=S2=S3=0.
      ONE=1.
      IF (IF2.LT.0) ONE=-1.
C
C
C     OUTPUT 'MK 17',IST1,IST2
      DO 200 I=NPC1,NPC2
      R1=OM1(IST1+I-1)
      C1=OM1(IST1+NPIECES+I-1)
```

```
      IF ((IF1.EQ.0).OR.(IF1.EQ.LPHALF))  C1=0.
      IF ((LFW1.EQ.LFW2).AND.(NSERIES.EQ.2).AND.
     $(ISQ.NE.3).AND.(ISQ.NE.4)) GO TO 180
      R2=OM2(IST2+I-1)
      C2=OM2(IST2+NPIECES+I-1)
      GO TO 190
  180 CONTINUE
      R2=OM1(IST2+I-1) ; C2=OM1(IST2+NPIECES+I-1)
  190 CONTINUE
      IF ((IF2A.EQ.0).OR.(IF2A.EQ.LPHALF))  C2=0.
      R3=W3(I)
      C3=W3(I+NPIECES)
      IF ((IF3A.EQ.0).OR.(IF3A.EQ.LPHALF))  C3=0.
      IF (INORM.EQ.0) GO TO 195
      BRP=        R1*R2*R3+ONE*R1*C2*C3+C1*R2*C3-ONE*C1*C2*R3
      BCP=  ONE*R1*C2*R3-     R1*R2*C3+C1*R2*R3+ONE*C1*C2*C3
      BMOD=DSQRT(BRP**2+BCP**2)
      BR=BR+BRP/BMOD ; BC=BC+BCP/BMOD
      GO TO 200
  195 CONTINUE
      BR=BR+      R1*R2*R3+ONE*R1*C2*C3+C1*R2*C3-ONE*C1*C2*R3
      BC=BC+ONE*R1*C2*R3-     R1*R2*C3+C1*R2*R3+ONE*C1*C2*C3
      S1=S1+R1*R1+C1*C1
      S2=S2+R2*R2+C2*C2
      S3=S3+R3*R3+C3*C3
  200 CONTINUE
C
C
      IF (INORM.NE.0) GO TO 204
  202 CONTINUE
      BR=BR/NPCS ; BC=BC/NPCS
      S1=S1/NPCS ; S2=S2/NPCS ; S3=S3/NPCS
C     OUTPUT S1,S2,S3,NPCS
      SD=DSQRT(S1*S2*S3)
C     OUTPUT BR,BC,SD
      BICOHR=BR/SD
      BICOHC=BC/SD
  204 CONTINUE
      BIPHSE=DATAN2(BC,BR)
      BIPHSE=BIPHSE*360./TWOPI
      IF (INORM.EQ.0) GO TO 206
      BIMOD=DSQRT(BR**2+BC**2)
      BICOH=BIMOD/NPCS
      IF (IF2.LT.0) GO TO 205
      BRU(IF3A)=BRU(IF3A)+BR/NPCS
      BIU(IF3A)=BIU(IF3A)+BC/NPCS
      BICU(IF3A)=BICU(IF3A)+BICOH
      BIC2U(IF3A)=BIC2U(IF3A)+BICOH**2
      BCSU(IF3A)=0.
      GO TO 208
  205 CONTINUE
      BRL(IF3A)=BRL(IF3A)+BR/NPCS
      BIL(IF3A)=BIL(IF3A)+BC/NPCS
      BICL(IF3A)=BICL(IF3A)+BICOH
      BIC2L(IF3A)=BIC2L(IF3A)+BICOH**2
      BCSL(IF3A)=0.
      GO TO 208
  206 CONTINUE
      BCM=BICOHR*BICOHR+BICOHC*BICOHC
      BICOH=SQRT(BCM)
```

```
C      OUTPUT 'TWO',MTOT
C      OUTPUT BICOH
       BM=BR*BR+BC*BC
       BIMOD=DSQRT(BM)
       IF (IF2.LT.0) GO TO 207
       BRU(IF3A)=BRU(IF3A)+BR
       BIU(IF3A)=BIU(IF3A)+BC
       BICU(IF3A)=BICU(IF3A)+BICOH
       BIC2U(IF3A)=BIC2U(IF3A)+BCM
       BCSU(IF3A)=BCSU(IF3A)+BCM*S3
       GO TO 208
   207 CONTINUE
       BRL(IF3A)=BRL(IF3A)+BR
       BIL(IF3A)=BIL(IF3A)+BC
       BICL(IF3A)=BICL(IF3A)+BICOH
       BIC2L(IF3A)=BIC2L(IF3A)+BCM
       BCSL(IF3A)=BCSL(IF3A)+BCM*S3
   208 CONTINUE
       IF (LISTBI.EQ.0) GO TO 215
       IF (INORM.NE.0) GO TO 212
       IF ((BICOH.LE.CONF1).OR.(BICOH.GT.CONF2)) GO TO 215
       IF (IHANN.NE.1) GO TO 210
       IF (BICOH.GE.SIGLVL) NPH=1 ;
      $PHSERR(1)=57.296*ASIN(1.96/SQRT(EDOF)/BICOH) ; GO TO 211
       NPH=0 ; GO TO 211
   210 CONTINUE
       IF ((IHANN.EQ.0).AND.(BICOH.GE.SIGLVL)) NPH=1 ;
      $PHSERR(1)=SQRT(2.*NPCS) ; GO TO 211
       NPH=0
   211 CONTINUE
       WRITE (108,2000) IF1,IF2,IF3,BICOH,BIPHSE,BIMOD,BR,BC,
      $S1,S2,S3,SD,NPH,(PHSERR(I),I=1,NPH)
  2000 FORMAT (X,I4,2(2X,I4),3X,F5.3,3X,F6.0,-X,1H ,3(2X,E11.3),
      $3(1X,E11.3),2X,E11.3,3X,N(F5.1))
       GO TO 215
   212 CONTINUE
       IF ((BICOH.LE.CONF1).OR.(BICOH.GT.CONF2)) GO TO 215
       IF (IHANN.NE.1) GO TO 213
       IF (BICOH.GE.SIGLVL) NPH=1 ;
      $PHSERR(1)=57.296*ASIN(1.96/SQRT(EDOF)/BICOH) ; GO TO 214
       NPH=0 ; GO TO 214
   213 CONTINUE
       IF ((IHANN.EQ.0).AND.(BICOH.GE.SIGLVL)) NPH=1 ;
      $PHSERR(1)=SQRT(2.*NPCS) ; GO TO 214
       NPH=0
   214 CONTINUE
       WRITE (108,2100) IF1,IF2,IF3,BICOH,BIPHSE,
      $BR/NPIECES,BC/NPIECES,NPH,(PHSERR(I),I=1,NPH)
  2100 FORMAT (X,I4,2(2X,I4),3X,F5.3,3X,F6.0,-X,1H ,
      $2(3X,E11.3),3X,N(F5.1))
   215 CONTINUE
C      OUTPUT IF1
C COMMIT BICOH TO STORAGE BUFFER SQ
       IF (IDUMP.EQ.0) GO TO 242
       CALL INDEX
       SQ(NROW,NCOL)=BICOH
   242 CONTINUE
C INCREMENT FREQUENCIES
       IF1=IF1+1     ; IF2=IF2-1
C END OF SCAN?
```

```
      CALL BOUNDARY
      GO TO (110,250),IBDY+1
C INCREMENT SCAN
  250 NCPT=NCPT+1 ; M33=0
C     WRITE (108,2222) IF1,ISQ,NCPT
C2222 FORMAT (1H ,'IF1=',I3,2X,'ISQ=',I2,2X,'NCPT=',I3)
C END OF RASTER?
      IF (NCPT.LE.NLAST) GO TO 105
C OUTPUT TO DISC
      IF (IDUMP.EQ.0) GO TO 260
      WRITE (ISTORE) ((SQ(I,J),I=1,NFHALF),J=1,NFHALF)
  260 CONTINUE
C INCREMENT INDEX FOR AREA BEING CONSIDERED (ISQ)
      IF (ISQOUT.NE.0) OUTPUT ISQ
      ISQ=ISQ+1
      IF ((NSERIES.EQ.1).AND.(ISQ.EQ.2)) ISQ=ISQ+1 ; GO TO 30
      IF ((NSERIES.EQ.1).AND.(ISQ.EQ.4)) GO TO 400
      IF (ISQ.LE.5) GO TO 30
C
  400 CONTINUE
C TRANSFER FOURIER LABELS TO BICOHERENCE FILE IF APPROPRIATE
      IF (IDUMP.EQ.0) GO TO 450
      IF (IDATSTART.NE.5) GO TO 420
      CALL RDISC(LFW1,1,LABL,4)
      WRITE (ISTORE) (LABL(I),I=1,4)
      CALL RDISC(LFW2,1,LABL,4)
      WRITE (ISTORE) (LABL(I),I=1,4)
      CALL RDISC(LFW1P2,1,LABL,4)
      WRITE (ISTORE) (LABL(I),I=1,4)
      WRITE (ISTORE) MF
      GO TO 440
  420 CONTINUE
      DO 425 I=1,3
  425 LABL(I)=4H
      LABL(4)=0
      DO 430 KZ=1,3
  430 WRITE (ISTORE) (LABL(I),I=1,4)
      WRITE (ISTORE) MF
  440 CONTINUE
      IF (MF.NE.21) GO TO 450
      CALL RDISC(21,1,LABL,33)
      WRITE (ISTORE) LABL
  450 CONTINUE
C
      CALL PRINTOUT
C
      GO TO 750
  700 STOP 'ABORTSET TERMINATION'
  750 STOP 'NORMAL PROGRAM COMPLETION'
C
C
      SUBROUTINE BOUNDARY
C INTERNAL SUBROUTINE FOR PROGRAM BISCAL03 TO TEST
C SCAN FOR FREQUENCY LIMITS
      IBDY=1
      GO TO (100,200,300,400,500),ISQ
  100 IF((IF1.LE.NFHALF).AND.(IF2.GT.0)) IBDY=0
      RETURN
  200 IF (IF1.LE.NFHALF) IBDY=0
      RETURN
```

```
  300 IF (IF2.GT.0) IBDY=0
      RETURN
  400 IF ((NCPT.LE.NFHALF).AND.(IF2.GE.-NFHALF)) IBDY=0
      IF ((NCPT.GT.NFHALF).AND.(IF1.LE.LPHALF)) IBDY=0
      RETURN
  500 IF (IF1.LE.LPHALF) IBDY=0
      RETURN
C
C
      SUBROUTINE FREQINIT
C INTERNAL SUBROUTINE FOR PROGRAM BISCAL03 TO INITIALIZE
C FREQUENCIES FOR A GIVEN LINE OF SCAN
      GO TO (100,200,300,400,500),ISQ
  100 IF1=NCPT/2 ; IF2=NCPT/2
      IF (MOD(NCPT,2).EQ.1) IF1=IF1+1
      RETURN
  200 IF1=NCPT+1 ; IF2=-1
      RETURN
  300 IF1=NFIRST-1 ; IF2=NCPT-IF1
      RETURN
  400 CONTINUE
      IF (NCPT.LE.NFHALF) IF1=NFHALF+1 ;
     $IF2=-IF1+NCPT
      IF (NCPT.GT.NFHALF) IF1=NCPT+1 ; IF2=-1
      RETURN
  500 IF1=NCPT+NFHALF+1 ; IF2=-NFHALF-1
      RETURN
C
C
      SUBROUTINE INDEX
C INTERNAL SUBROUTINE FOR PROGRAM BISCAL03 TO RELATE
C FREQUENCIES TO STORAGE ARRAY INDICES
      GO TO (100,200,300,400,500),ISQ
  100 NROW=IF2 ; NCOL=IF1
      RETURN
  200 NROW=IF2+NFHALF+1 ; NCOL=IF1
      RETURN
  300 NROW=IF2 ; NCOL=IF1-NFHALF
      RETURN
  400 NROW=IF2+NFHALF+1 ; NCOL=IF1-NFHALF
      RETURN
  500 NROW=IF2+2*NFHALF+1 ; NCOL=IF1-NFHALF
      RETURN
C
C
      SUBROUTINE RASTER
C INTERNAL SUBROUTINE FOR PROGRAM BISCAL03 TO SET UP SCAN
C PARAMETERS
      GO TO (100,200,300,400,500),ISQ
  100 NFIRST=2 ; NLAST=LPHALF
      RETURN
  200 NFIRST=1 ; NLAST=NFHALF-1
      RETURN
  300 NFIRST=NFHALF+2 ; NLAST=LPHALF
      RETURN
  400 NFIRST=1 ; NLAST=LPHALF-1
      RETURN
  500 NFIRST=1 ; NLAST=NFHALF-1
      RETURN
C
```

```
C
      SUBROUTINE SUMFREQ
C INTERNAL SUBROUTINE FOR PROGRAM BISCAL03 TO DETERMINE WHERE
C TO FIND SUM FREQUENCY COEFFICIENTS
C RETURNS IREAD=1 IF READING IS NECESSARY, 0 OTHERWISE
      IREAD=0
      GO TO (100,400,150,200,300),ISQ
  100 IF (IF3A.GT.NFHALF) IREAD=1
      RETURN
  150 CONTINUE
      IF (NSERIES.EQ.1) RETURN
      IF (LFW1.EQ.LFW2) IREAD=1 ; RETURN
      IF ((LFW1.NE.LFW2).AND.(LFW1P2.EQ.LFW2)) IREAD=1 ; RETURN
      RETURN
  200 CONTINUE
      IF ((LFW1P2.EQ.LFW1).AND.(IF3A.LE.NFHALF).AND.
     $(LFW1.NE.LFW2)) IREAD=1 ; RETURN
      IF ((LFW1P2.EQ.LFW2).AND.(IF3A.GT.NFHALF).AND.
     $(LFW1.NE.LFW2)) IREAD=1 ; RETURN
      IF ((LFW1.EQ.LFW2).AND.(NSERIES.EQ.2)) IREAD=1 ; RETURN
      RETURN
  300 IF (LFW1.NE.LFW2) IREAD=1
  400 CONTINUE
      RETURN
C
C
      SUBROUTINE PRINTOUT
C INTERNAL SUBROUTINE FOR PROGRAM BISUM01 TO PRINT BISPECTRAL
C INTEGRALS
      IF (KDISC.EQ.0) GO TO 70
      IF (NSERIES.EQ.1) GO TO 50
      WRITE (KDCB,5000) 2*(LPHALF-1)
 5000 FORMAT (' CROSS SPECTRUM *** TOTAL NUMBER OF SUM ',
     $'FREQ''S: ',I4,' *** EACH BINARY RECORD EQUALS ONE LINE '
     $,'OF PRINTED OUTPUT')
      GO TO 70
   50 WRITE (KDCB,6000) LPHALF-1
 6000 FORMAT ('  AUTO SPECTRUM *** TOTAL NUMBER OF SUM ',
     $'FREQ''S: ',I4,' *** EACH BINARY RECORD EQUALS ONE LINE '
     $,'OF PRINTED OUTPUT')
   70 CONTINUE
      NLINES=0
      WRITE (108,1000)
 1000 FORMAT (1H1,T26,'I N T E G R A L S   O V E R   ',
     $'P O S I T I V E   F R E Q U E N C I E S   A L O N E',/)
      NLINES=NLINES+2
      WRITE (108,1200)
 1200 FORMAT (X,'SUM   NO IN',T127,'SUM',/,X,'FREQ   SCAN',
     $T18,'BR',T26,'BR/N',T39,'BI',T48,'BI/N',T60,'BIC',T70,
     $'BIC/N',T82,'BIC**2  (BIC**2)/N',T104,'S(W3)*   (S(W3)*',
     $T127,'FREQ',/,'SEQ NO  (N)',T106,
     $'BIC**2   BIC**2)/N  SEQ NO',/)
      NLINES=NLINES+4
C
C
      DO 100 I=2,LPHALF
      NLINES=NLINES+1
      IF (NLINES.GT.60) NLINES=6 ; WRITE (108,2000) ;
     $WRITE (108,1200)
 2000 FORMAT (1H1)
```

```
      KP=I/2
      WRITE (108,1500) I,KP,BRU(I),BRU(I)/KP,BIU(I),BIU(I)/KP,
     $BICU(I),BICU(I)/KP,BIC2U(I),BIC2U(I)/KP,
     $BCSU(I),BCSU(I)/KP,I
 1500 FORMAT (X,I4,T9,I3,T14,3(G9.3,X,G9.3,3X),X,
     $2(G9.3,X,G9.3,4X),I4)
      IF (KDISC.NE.0)
     $WRITE (KDCB)      I,KP,BRU(I),BRU(I)/KP,BIU(I),BIU(I)/KP,
     $BICU(I),BICU(I)/KP,BIC2U(I),BIC2U(I)/KP,
     $BCSU(I),BCSU(I)/KP,I
  100 CONTINUE
C
C
      IF (NSERIES.EQ.1) GO TO 300
      NLINES=0
      WRITE (108,4000)
 4000 FORMAT(1H1,T27,'I N T E G R A L S   I N V O L V I N G   '
     $,'N E G A T I V E   F R E Q U E N C I E S',/)
      NLINES=NLINES+2
      WRITE (108,1200)
      NLINES=NLINES+4
C
C
      DO 200 I=1,LPHALF-1
      NLINES=NLINES+1
      IF (NLINES.GT.60) NLINES=6 ; WRITE (108,2000) ;
     $WRITE (108,1200)
      KP=LPHALF-I
      WRITE (108,1500) I,KP,BRL(I),BRL(I)/KP,BIL(I),BIL(I)/KP,
     $BICL(I),BICL(I)/KP,BIC2L(I),BIC2L(I)/KP,
     $BCSL(I),BCSL(I)/KP,I
      IF (KDISC.NE.0)
     $WRITE (KDCB)      I,KP,BRL(I),BRL(I)/KP,BIL(I),BIL(I)/KP,
     $BICL(I),BICL(I)/KP,BIC2L(I),BIC2L(I)/KP,
     $BCSL(I),BCSL(I)/KP,I
  200 CONTINUE
C
C
  300 CONTINUE
      RETURN
      END
```

- PROGRAM BIVEC -

```
C                   <<<<<<<  B I V E C 0 5  >>>>>>>
C CALCULATES ROTARY AUTO-BISPECTRUM AND ROTARY CROSS-BISPECTRUM
C FOR ONE OR TWO VECTOR SERIES AND DETERMINES CONFIDENCE LEVELS
C FOR BICOHERENCES.
C GENERATES TWO OR FOUR DISC FILES FOR TRANSMITTAL TO
C PLOTTING PROGRAM.
C PROGRAMMER:  GERARD H. MARTINEAU
C ORIGINATOR:  MELBOURNE G. BRISCOE
C DATE       :  OCT 17, 1977
C*************************************************************
      DIMENSION W11(2560),W12(2560),W21(2560),W22(2560),
     $OM(2560,4),SQ1(35,35),SQ2(35,35),W31(80),W32(80),
     $KBIN(200),ABIN(200),CONF(10),BCL(10),KARD(20),LW(6)
      DIMENSION LABL(33),PHSERR(2)
      EQUIVALENCE (W11,OM),(W12,OM(1,2)),(W21,OM(1,3)),
     $(W22,OM(1,4)),(KBIN,ABIN),
     $(LW(1),LW11),(LW(2),LW12),(LW(3),LW21),(LW(4),LW22),
     $(LW(5),LW31),(LW(6),LW32)
      DOUBLE PRECISION A11,B11,A12,B12,A21,B21,A22,B22,A31,B31,
     $A32,B32,P1U,P2U,P3U,P1V,P2V,P3V,
     $RNUMU,RNUMV,RDENU,RDENV,RBUR,RBUI,RBVR,RBVI,
     $U1R,U1I,U2R,U2I,U3R,U3I,V1R,V1I,V2R,V2I,V3R,V3I,RNPCS,
     $RMODU,RMODV,RBURT,RBUIT,RBVRT,RBVIT,ANPCS
      COMMON /PRIME/ NPRIME(65)
      DATA NPRIME/
     $  16,  24,  32,  40,  48,  64,  72,  80,  96, 120,
     $ 128, 144, 160, 200, 216, 240, 256, 288, 320, 360,
     $ 384, 400, 432, 480, 512, 576, 600, 640, 648, 720,
     $ 768, 800, 864, 960,1024,1080,1152,1200,1280,1296,
     $1440,1536,1600,1728,1800,1920,1944,2000,2048,2160,
     $2304,2400,2560,2592,2880,2916,3000,3072,3200,3240,
     $3456,3600,3848,3888,4000/
      NAMELIST IDATSTART,MF,ISTORE1,ISTORE2,BINSIZE,RBICLIM,
     $IHI,KB1,KB2,KONF,LISTBI,IDUMP,LABREAD,IFOURIER,INORM,
     $ISTORE3,ISTORE4,ISQOUT,IHANN,SIGLVL
      ISQD=35 ; ISOM=2560
      PI=3.14159265 ; TWOPI=2.*PI
      DTR=PI/180. ; RTD=180./PI
      ISQOUT=0 ; IHANN=1
      SIGLVL=-1 ; PHSERR(2)=0
      CALL ABORTSET(700S,1)
      CALL BIVCTRL
      IF ((LISTBI.NE.0).AND.(INORM.EQ.0)) WRITE (108,1321)
 1321 FORMAT (/,T3,'F1',T9,'F2',T15,'F3',T20,'RBICOH  RBIPHASE'
     $,T39,'/<RB>/',T51,'<REAL(RB)>',T64,'<IMAG(RB)>',T76,
     $'<P(W1)DW>',T88,'<P(W2)DW>',T100,'<P(W3)DW>',T111,
     $'SQRT(PROD<P>)',T125,'PHSERR',/)
      IF ((LISTBI.NE.0).AND.(INORM.NE.0)) WRITE (108,1431)
 1431 FORMAT (/,T3,'F1',T9,'F2',T15,'F3',T20,'RBICOH  RBIPHASE'
     $,T38,'<REAL(RB'')>',T52,'<IMAG(RB'')>',T65,'PHSERR',/)
      NPCS=NPC2-NPC1+1
      IF ((NPCS.EQ.1).AND.(IHANN.EQ.1)) IHANN=0 ; OUTPUT
     $'IHANN HAS BEEN SET TO 0. CAN''T OVERLAP WITH ONE PIECE'
      IF (IHANN.EQ.1) EDOF=36.*NPCS*NPCS/(19*NPCS-1)
      IF (IHANN.EQ.0) EDOF=2*NPCS
      IF (SIGLVL.LT.0) SIGLVL=SQRT(6./EDOF)
      LPATHE=LPIECE/8 ; LPQU=LPIECE/4 ; LPHALF=LPIECE/2
      LP38=3*LPATHE ; IQNO=2*NPIECES*LPATHE
```

```
      IQU1=IDATSTART+NPIECES ; IQU2=IQU1+IQNO
      IQU3=IQU1+2*IQNO ; IQU4=IQU1+3*IQNO
C START OF LOOP FOR /W2/>/W1/
      KPASS=1
   50 CONTINUE
C INITIALIZE RASTER AT ZERO
      ISQ=0
C
C
C STARTING POINT FOR RASTER LOOP:
  100 CONTINUE
C OUTPUT TO DISC
      IF ((IDUMP.EQ.0).OR.(ISQ.EQ.0)) GO TO 102
      KST1=ISTORE1 ; KST2=ISTORE2
      IF (KPASS.EQ.2) KST1=ISTORE3 ; KST2=ISTORE4
      WRITE (KST1)    ((SQ1(I,J),I=1,LPATHE),J=1,LPATHE)
      WRITE (KST2)    ((SQ2(I,J),I=1,LPATHE),J=1,LPATHE)
  102 CONTINUE
C INCREMENT RASTER
      ISQ=ISQ+1
      IF (ISQOUT.NE.0) OUTPUT ISQ
C IS RASTER LIMIT EXCEEDED?
      IF (ISQ.GT.16) GO TO 300
C SET STORAGE ARRAYS SQ1,SQ2 TO -999. IF DUMPING
      IF (IDUMP.EQ.0) GO TO 110
C
C
      DO 105 I=1,ISQD
C
      DO 105 J=1,ISQD
      SQ1(I,J)=-999.
  105 SQ2(I,J)=-999.
C
C
  110 CONTINUE
      CALL FREQBIAS12
      IF ((ISQ.NE.11).AND.(ISQ.NE.16)) GO TO 125
C
C
      DO 120 I=1,ISOM
      W11(I)=0.
  120 W12(I)=0.
C
C
  125 IF ((ISQ.NE.16).OR.(ISW.NE.3)) GO TO 135
C
C
      DO 130 I=1,ISOM
      W21(I)=0.
  130 W22(I)=0.
C
C
  135 CONTINUE
      IF (KPASS.EQ.1) GO TO 137
      LWP=LW11 ; LWQ=LW12
      LW11=LW21 ; LW12=LW22 ; LW21=LWP ; LW22=LWQ
  137 CALL READW1W2
      IF (KPASS.EQ.1) GO TO 138
      LWP=LW11 ; LWQ=LW12
      LW11=LW21 ; LW12=LW22 ; LW21=LWP ; LW22=LWQ
```

```
X       WRITE (108,1717) ((OM(I,J),I=1,4),J=1,4)
   138 CALL RASTER
C  INITIALIZE SCAN (SUM) FREQUENCY
        NCPT=IFIRST-1
C
C
C  STARTING POINT FOR SCAN LOOP:
   140 CONTINUE
C  INCREMENT SCAN
        NCPT=NCPT+1
X       OUTPUT NCPT
C  CHECK FOR END OF RASTER
        IF (NCPT.GT.ILAST) GO TO 100
        IF (KPASS.EQ.1) GO TO 145
        LWP=LW11 ; LWQ=LW12
        LW11=LW21 ; LW12=LW22 ; LW21=LWP ; LW22=LWQ
   145 CALL READW3
        IF (KPASS.EQ.1) GO TO 150
        LWP=LW11 ; LWQ=LW12
        LW11=LW21 ; LW12=LW22 ; LW21=LWP ; LW22=LWQ
X       WRITE (108,7777) (W31(N),N=1,2),(W32(N),N=1,2)
X7777 FORMAT (X,2(2F2.0,4X))
   150 CALL FREQINIT
C  INITIALIZE FREQUENCIES 1,2
        IF1=IF1-1 ; IF2=IF2+1
C
C
C
C  STARTING POINT FOR FREQ. 1,2 LOOP (GRID POINT LOOP)
   155 CONTINUE
C  INCREMENT FREQUENCIES 1,2
        IF1=IF1+1 ; IF2=IF2-1
        IF2ABS=ABS(IF2)
C  CHECK FOR END OF SCAN
        CALL BOUNDARY
        IF (IBDY.EQ.1) GO TO 140
C  BEGIN CALCULATIONS
        RBUR=RBUI=RBVR=RBVI=0.
        P1U=P2U=P3U=P1V=P2V=P3V=0.
C
C
        DO 200 I=NPC1,NPC2
        IF3=IF1+IF2
        IA1=(IF1-IFB1-1)*2*NPIECES+I
        IB1=IA1+NPIECES
        IA2=(IF2ABS-IFB2-1)*2*NPIECES+I
        IB2=IA2+NPIECES
        A11=W11(IA1) ; B11=W11(IB1)
        A12=W12(IA1) ; B12=W12(IB1)
        IF (ISW.EQ.3) GO TO 170
        GO TO (160,160,170,170,160,160,170,170,170,170,170,170,
       $170,170,160,160),ISQ
   160 A21=W11(IA2) ; B21=W11(IB2)
        A22=W12(IA2) ; B22=W12(IB2)
        GO TO 180
   170 A21=W21(IA2) ; B21=W21(IB2)
        A22=W22(IA2) ; B22=W22(IB2)
   180 A31=W31(I) ; B31=W31(NPIECES+I)
        A32=W32(I) ; B32=W32(NPIECES+I)
X       WRITE (108,1717) A11,A12,B11,B12,A21,A22,B21,B22,
```

```
     X      $A31,A32,B31,B32
     X1717 FORMAT (X,3(4F4.0,4X))
            U1R=A11+B12 ; U1I=A12-B11
            U3R=A31+B32 ; U3I=-A32+B31
            V1R=A11-B12 ; V1I=A12+B11
            V3R=A31-B32 ; V3I=-A32-B31
            IF (IF2.LT.0) GO TO 185
            U2R=A21+B22 ; U2I=A22-B21
            V2R=A21-B22 ; V2I=A22+B21
            P2U=P2U+U2R**2+U2I**2 ; P2V=P2V+V2R**2+V2I**2
            GO TO 190
        185 U2R=A21-B22 ; U2I=A22+B21
            V2R=A21+B22 ; V2I=A22-B21
            P2U=P2U+U2R**2+U2I**2 ; P2V=P2V+V2R**2+V2I**2
        190 CONTINUE
            IF (INORM.EQ.0) GO TO 195
            RBURT=U1R*U2R*U3R-U1I*U2I*U3R-U1R*U2I*U3I-U1I*U2R*U3I
            RBUIT=U1R*U2I*U3R+U1I*U2R*U3R+U1R*U2R*U3I-U1I*U2I*U3I
            RBVRT=V1R*V2R*V3R-V1I*V2I*V3R-V1R*V2I*V3I-V1I*V2R*V3I
            RBVIT=V1R*V2I*V3R+V1I*V2R*V3R+V1R*V2R*V3I-V1I*V2I*V3I
            RMODU=DSQRT(RBURT**2+RBUIT**2)
            RBUR=RBUR+RBURT/RMODU ; RBUI=RBUI+RBUIT/RMODU
            RMODV=DSQRT(RBVRT**2+RBVIT**2)
            RBVR=RBVR+RBVRT/RMODV ; RBVI=RBVI+RBVIT/RMODV
            GO TO 200
        195 CONTINUE
            P1U=P1U+U1R**2+U1I**2 ; P3U=P3U+U3R**2+U3I**2
            P1V=P1V+V1R**2+V1I**2 ; P3V=P3V+V3R**2+V3I**2
            RBUR=RBUR+U1R*U2R*U3R-U1I*U2I*U3R-U1R*U2I*U3I-U1I*U2R*U3I
            RBUI=RBUI+U1R*U2I*U3R+U1I*U2R*U3R+U1R*U2R*U3I-U1I*U2I*U3I
            RBVR=RBVR+V1R*V2R*V3R-V1I*V2I*V3R-V1R*V2I*V3I-V1I*V2R*V3I
            RBVI=RBVI+V1R*V2I*V3R+V1I*V2R*V3R+V1R*V2R*V3I-V1I*V2I*V3I
        200 CONTINUE
     C
     C
            IF (INORM.EQ.0) GO TO 202
            RBICU=(DSQRT(RBUR**2+RBUI**2))/ANPCS
            RBICV=(DSQRT(RBVR**2+RBVI**2))/ANPCS
            GO TO 204
        202 CONTINUE
            RNUMU=DSQRT(RBUR**2+RBUI**2) ; RDENU=DSQRT(P1U*P2U*P3U)
            RBICU=RNPCS*RNUMU/RDENU
            RNUMV=DSQRT(RBVR**2+RBVI**2) ; RDENV=DSQRT(P1V*P2V*P3V)
            RBICV=RNPCS*RNUMV/RDENV
        204 CONTINUE
            RBPHU=DATAN(RBUI,RBUR)
            RBPHV=DATAN(RBVI,RBVR)
            RBPHU=RTD*RBPHU ; RBPHV=RTD*RBPHV
     C SORT ROT BIC'S IF CALLED FOR
            IF (KONF.NE.0) CALL CBIN(RBICU,1) ; CALL CBIN(RBICV,2)
     C PRINT ROT BIC'S WITHIN SPECIFIED LIMITS
            IF (LISTBI.EQ.0) GO TO 219
            JF1=IF1 ; JF2=IF2
            IF (KPASS.EQ.2) JF1=IF2 ; JF2=IF1
            IF (INORM.NE.0) GO TO 212
            IF ((RBICU.LE.CONF1).OR.(RBICU.GT.CONF2)) GO TO 208
            IF (IHANN.NE.1) GO TO 205
            IF (RBICU.GE.SIGLVL) NPH=1 ;
           $PHSERR(1)=57.296*ASIN(1.96/SQRT(EDOF)/RBICU) ; GO TO 206
            NPH=0 ; GO TO 206
```

```
  205 CONTINUE
      IF ((IHANN.EQ.0).AND.(RBICU.GE.SIGLVL)) NPH=1 ;
     $PHSERR(1)=SQRT(2.*NPCS) ; GO TO 206
      NPH=0
  206 CONTINUE
      WRITE (108,2000) JF1,JF2,IF3,RBICU,RBPHU,RNUMU/NPIECES,
     $RBUR/NPIECES,RBUI/NPIECES,P1U/NPIECES,P2U/NPIECES,
     $P3U/NPIECES,RDENU/NPIECES**1.5,NPH,(PHSERR(I),I=1,NPH)
 2000 FORMAT (X,I4,2(2X,I4),3X,F5.3,3X,F6.0,-X,1H  ,3(2X,E11.3)
     $,3(1X,E11.3),2X,E11.3,3X,N(F5.1))
  208 CONTINUE
      IF ((RBICV.LE.CONF1).OR.(RBICV.GT.CONF2)) GO TO 219
      IF (IHANN.NE.1) GO TO 209
      IF (RBICV.GE.SIGLVL)  NPH=1 ;
     $PHSERR(1)=57.296*ASIN(1.96/SQRT(EDOF)/RBICV) ; GO TO 210
      NPH=0 ; GO TO 210
  209 CONTINUE
      IF ((IHANN.EQ.0).AND.(RBICV.GE.SIGLVL)) NPH=1 ;
     $PHSERR(1)=SQRT(2.*NPCS) ; GO TO 210
      NPH=0
  210 CONTINUE
      WRITE (108,2000) -JF1,-JF2,-IF3,RBICV,RBPHV,RNUMV/NPIECES
     $,RBVR/NPIECES,RBVI/NPIECES,P1V/NPIECES,P2V/NPIECES,
     $P3V/NPIECES,RDENV/NPIECES**1.5,NPH,(PHSERR(I),I=1,NPH)
      GO TO 219
  212 CONTINUE
      IF ((RBICU.LE.CONF1).OR.(RBICU.GT.CONF2)) GO TO 216
      IF (IHANN.NE.1) GO TO 213
      IF (RBICU.GE.SIGLVL) NPH=1 ;
     $PHSERR(1)=57.296*ASIN(1.96/SQRT(EDOF)/RBICU) ; GO TO 214
      NPH=0 ; GO TO 214
  213 CONTINUE
      IF ((IHANN.EQ.0).AND.(RBICU.GE.SIGLVL)) NPH=1 ;
     $PHSERR(1)=SQRT(2.*NPCS) ; GO TO 214
      NPH=0.
  214 CONTINUE
      WRITE (108,2100) JF1,JF2,IF3,RBICU,RBPHU,RBUR/NPIECES,
     $RBUI/NPIECES,NPH,(PHSERR(I),I=1,NPH)
 2100 FORMAT(X,I4,2(2X,I4),3X,F5.3,3X,F6.0,-X,1H ,
     $2(3X,E11.3),3X,N(F5.1))
  216 CONTINUE
      IF ((RBICV.LE.CONF1).OR.(RBICV.GT.CONF2)) GO TO 219
      IF (IHANN.NE.1) GO TO 217
      IF (RBICV.GE.SIGLVL) NPH=1 ;
     $PHSERR(1)=57.296*ASIN(1.96/SQRT(EDOF)/RBICV) ; GO TO 218
      NPH=0 ; GO TO 218
  217 CONTINUE
      IF ((IHANN.EQ.0).AND.(RBICV.GE.SIGLVL)) NPH=1 ;
     $PHSERR(1)=SQRT(2.*NPCS) ; GO TO 218
      NPH=0
  218 CONTINUE
      WRITE (108,2200) -JF1,-JF2,-IF3,RBICV,RBPHV,RBVR/NPIECES,
     $RBVI/NPIECES,NPH,(PHSERR(I),I=1,NPH)
 2200 FORMAT (X,I4,2(2X,I4),3X,F5.3,3X,F6.0,-X,1H ,
     $2(3X,E11.3),3X,N(F5.1))
  219 CONTINUE
C COMMIT RBICU TO SQ1 AND RBICV TO SQ2
      IF (IDUMP.EQ.0) GO TO 240
      NC1=IF1-IFB1
      IF (IF2.LT.0) NR1=IF2+1+LPATHE+IFB2 ; GO TO 220
```

```
      NR1=IF2-IFB2
  220 SQ1(NR1,NC1)=RBICU
      NC2=IFB1+LPATHE-IF1+1
      IF (IF2.LT.0) NR2=-IFB2-IF2 ; GO TO 230
      NR2=IFB2+LPATHE-IF2+1
  230 SQ2(NR2,NC2)=RBICV
  240 CONTINUE
C
C END OF FREQ. 1,2 LOOP (GRID POINT LOOP)
      GO TO 155
C
C
  300 CONTINUE
C OUTPUT APPROPRIATE LABELS TO DISC
      IF (IDUMP.EQ.0) GO TO 400
      IF (LABREAD.EQ.0) GO TO 360
C
      DO 350 I=1,6
  350 CALL RDISC(LW(I),1,LABL(4*I-3),4)
C
  360 CONTINUE
C
C
      KST1=ISTORE1 ; KST2=ISTORE2
      IF (KPASS.EQ.2) KST1=ISTORE3 ; KST2=ISTORE4
      DO 370 I=1,6
      WRITE (KST1)    (LABL(K),K=4*I-3,4*I)
      WRITE (KST2)    (LABL(K),K=4*I-3,4*I)
  370 CONTINUE
C
C
      WRITE (KST1)    MF
      WRITE (KST2)    MF
C
      DO 380 I=1,33
  380 LABL(I)=4H
      IF ((MF.NE.21).OR.(IFOURIER.EQ.0)) GO TO 390
C
      CALL RDISC(21,1,LABL,33)
  390 CONTINUE
      WRITE (KST1)    LABL
      WRITE (KST2)    LABL
  400 CONTINUE
C END OF LOOP FOR /W2/>/W1/
      IF ((ISW.EQ.3).AND.(KPASS.EQ.1)) KPASS=2 ; GO TO 50
C DETERMINE CONFIDENCE LEVELS IF CALLED FOR
      IF (KONF.NE.0) CALL CONFIDENCE
C
      GO TO 800
  700 STOP 'ABORTSET TERMINATION'
C
  800 STOP 'NORMAL PROGRAM COMPLETION'
C**************************************************************
C**************************************************************
      SUBROUTINE BIVCTRL
C INTERNAL SUBROUTINE FOR PROGRAM BIVEC01 TO INITIALIZE,
C AND READ AND CHECK INPUT PARAMETERS.
      IDUMP=1
      MF=21 ; IDATSTART=5
      ISTORE1=1 ; ISTORE2=2 ; ISTORE3=3 ; ISTORE4=4
```

```
      IHI=1
      LABREAD=1 ; IFOURIER=1
      BINSIZE=0.01 ; RBICLIM=1.5 ; KB1=KB2=1 ; KONF=0
      KOVER=0 ; LISTBI=0
      MTOT=0 ; LIMPCS=150
      INORM=0
C
C
      DO 20 I=1,200
   20 KBIN(I)=0
C
C
      DO 40 I=1,33
   40 LABL(I)=4H
C
C
      INPUT
      OUTPUT IDATSTART,MF,ISTORE1,ISTORE2,ISTORE3,ISTORE4,
     $BINSIZE,RBICLIM,IHI,KB1,KB2,KONF,LISTBI,IDUMP,LABREAD,
     $IFOURIER,INORM,IHANN
      CALL DPAR(NERR,MF,1,1,KSUM,512)
      READ (105,1000) KARD
 1000 FORMAT (20A4)
      WRITE (108,1100) KARD
 1100 FORMAT (X,20A4)
      DECODE (80,1200,KARD) LW11,LW12,LW21,LW22,LW31,LW32,
     $NPIECES,LPIECE,NPC1,NPC2,CONF1,CONF2
 1200 FORMAT (12G)
      IF (NPC2.GT.NPIECES) OUTPUT NPC2,NPIECES,
     $'LAST PIECE TO AVG GREATER THAN NO OF PCS' ; STOP
      ANPCS=DFLOAT(NPIECES)
      RNPCS=DSQRT(ANPCS)
      IF ((LW11.NE.LW21).OR.(LW12.NE.LW22)) GO TO 30
      IF ((LW31.EQ.LW11).AND.(LW32.EQ.LW12)) ISW=1 ; GO TO 50
      ISW=2 ; GO TO 50
   30 CONTINUE
      IF (((LW31.EQ.LW11).AND.(LW32.EQ.LW12)).OR.
     $((LW31.EQ.LW21).AND.(LW32.EQ.LW22))) ISW=3 ; GO TO 50
      OUTPUT LW11,LW12,LW21,LW22,LW31,LW32,
     $'NO MORE THAN TWO DIFFERENT VECTOR SERIES ALLOWED' ; STOP
   50 CONTINUE
C CHECK FOR EXISTENCE OF DCB ASSIGNMENTS FOR OUTPUT FILES
      IF (IDUMP.EQ.0) GO TO 80
      N1=ISTORE1 ; N2=ISTORE2 ; N3=ISTORE3 ; N4=ISTORE4
      IF (ISW.EQ.3) GO TO 60
      IF (N1.NE.N2) GO TO 70
      OUTPUT ISTORE1,ISTORE2 ; GO TO 65
   60 CONTINUE
      IF ((N1.NE.N2).AND.(N1.NE.N3).AND.(N1.NE.N4).AND.
     $(N2.NE.N3).AND.(N2.NE.N4).AND.(N3.NE.N4)) GO TO 70
      OUTPUT ISTORE1,ISTORE2,ISTORE3,ISTORE4
   65 OUTPUT 'TWO OUTPUT DCB''S HAVE BEEN SET EQUAL CAN''T DO.'
     $ ; STOP
   70 CONTINUE
      CALL GETDCB(ISTORE1,LOCC)
      CALL GETDCB(ISTORE2,LOCC)
      IF (ISW.NE.3) GO TO 80
      CALL GETDCB(ISTORE3,LOCC)
      CALL GETDCB(ISTORE4,LOCC)
   80 CONTINUE
```

```
C
C
      DO 100 I=1,65
      IF (LPIECE.EQ.NPRIME(I)) GO TO 150
  100 CONTINUE
C
C
      WRITE (108,1250) NPRIME ; STOP
 1250 FORMAT ('PIECE LENGTH MUST BE FROM FOLLOWING LIST:',/,
     $(10(X,I7)))
  150 CONTINUE
      IF (NPIECES.GT.LIMPCS) WRITE (108,1275) ; STOP
 1275 FORMAT ('NUMBER OF PIECES MUST NOT EXCEED 150')
      IF (KONF.EQ.0) GO TO 300
      READ (105,1000) KARD
      WRITE (108,1100) KARD
      DECODE (80,3000,KARD) NCONF,NCONF,(CONF(I),I=1,NCONF)
 3000 FORMAT (G,NG)
      NBINS=INT(RBICLIM/BINSIZE+0.5)
      IF (NBINS.GT.200) WRITE (108,3050) ; STOP
 3050 FORMAT ('NUMBER OF BINS MUST NOT EXCEED 200')
      RBICLIM=NBINS*BINSIZE
      WRITE (108,4000) RBICLIM
 4000 FORMAT ('ADJUSTED ROT BICOH LIMIT, HOLDING BINSIZE ',
     $'AS INPUT ',F6.2)
  300 CONTINUE
      RETURN
C*******************************************************************
C*******************************************************************
      SUBROUTINE FREQBIAS12
C INTERNAL SUBROUTINE FOR PROGRAM BIVEC01 TO
C RETURN FREQUENCY BIASES FB1,FB2
      GO TO (10,10,30,30,50,50,70,70,90,90,110,110,130,140,
     $150,160),ISQ
   10 IFB1=IFB2=0. ; RETURN
   30 IFB1=LPATHE ; IFB2=0. ; RETURN
   50 IFB1=LPATHE ; IFB2=LPATHE ; RETURN
   70 IFB1=LPQU ; IFB2=LPATHE ; RETURN
   90 IFB1=LPQU ; IFB2=0. ; RETURN
  110 IFB1=LP38 ; IFB2=0. ; RETURN
  130 IFB1=LP38 ; IFB2=LPATHE ; RETURN
  140 IFB1=LP38 ; IFB2=LPQU ; RETURN
  150 IFB1=LPQU ; IFB2=LPQU ; RETURN
  160 IFB1=LP38 ; IFB2=LP38 ; RETURN
C*******************************************************************
C*******************************************************************
      SUBROUTINE READW1W2
C INTERNAL SUBROUTINE FOR PROGRAM BIVEC01
C TO FILL BUFFERS FOR FREQUENCIES 1,2
      GO TO (10,20,30,40,50,60,70,80,90,100,110,120,130,140,
     $150,160),ISQ
C
   10 DO 13 I=1,2
   13 CALL RDISC(LW(I),IQU1,OM(1,I),IQNO)
C
      DO 18 I=1,2
      GO TO (14,15,16),ISW
   14 CALL RDISC(LW(I),IQU2,OM(1,I+2),IQNO) ; GO TO 18
   15 CALL RDISC(LW(I+4),IQU1,OM(1,I+2),IQNO) ; GO TO 18
   16 CALL RDISC(LW(I+2),IQU1,OM(1,I+2),IQNO)
```

```
   18 CONTINUE
C
   20 CONTINUE
      IF (ISW.NE.2) GO TO 26
C
      DO 24 I=1,2
   24 CALL RDISC(LW(I+4),IQU1,OM(1,I+2),IQNO)
C
   26 RETURN
C
   30 DO 35 I=1,2
      CALL RDISC(LW(I),IQU2,OM(1,I),IQNO)
      IF (ISW.NE.3) CALL RDISC(LW(I+2),IQU1,OM(1,I+2),IQNO)
   35 CONTINUE
C
   40 RETURN
C
   50 DO 58 I=1,2
      GO TO (54,55,56),ISW
   54 CALL RDISC(LW(I),IQU3,OM(1,I+2),IQNO) ; GO TO 58
   55 CALL RDISC(LW(I+4),IQU3,OM(1,I+2),IQNO) ; GO TO 58
   56 CALL RDISC(LW(I+2),IQU2,OM(1,I+2),IQNO)
   58 CONTINUE
C
      RETURN
   60 IF (ISW.EQ.3) RETURN
C
      DO 68 I=1,2
      GO TO (64,65),ISW
   64 CALL RDISC(LW(I),IQU1,OM(1,I+2),IQNO) ; GO TO 68
   65 CALL RDISC(LW(I+4),IQU1,OM(1,I+2),IQNO)
   68 CONTINUE
C
      RETURN
C
   70 DO 73 I=1,2
   73 CALL RDISC(LW(I),IQU3,OM(1,I),IQNO)
C
      IF (ISW.EQ.3) RETURN
C
      DO 78 I=1,2
      CALL RDISC(LW(I+2),IQU2,OM(1,I+2),IQNO)
   78 CONTINUE
C
   80 RETURN
C
   90 DO 95 I=3,4
   95 CALL RDISC(LW(I),IQU1,OM(1,I),IQNO)
C
  100 RETURN
C
  110 DO 115 I=1,2
  115 CALL RDISC(LW(I),IQU4,OM(1,I),IQNO-NPIECES)
C
  120 RETURN
C
  130 DO 135 I=3,4
  135 CALL RDISC(LW(I),IQU2,OM(1,I),IQNO)
C
      RETURN
```

```
C
  140 DO 145 I=3,4
  145 CALL RDISC(LW(I),IQU3,OM(1,I),IQNO)
C
      RETURN
C
  150 DO 153 I=1,2
  153 CALL RDISC(LW(I),IQU3,OM(1,I),IQNO)
C
      IF (ISW.EQ.3) RETURN
C
      DO 158 I=1,2
      GO TO (154,155),ISW
  154 CALL RDISC(LW(I),IQU1,OM(1,I+2),IQNO) ; GO TO 158
  155 CALL RDISC(LW(I+4),IQU1,OM(1,I+2),IQNO)
  158 CONTINUE
C
      RETURN
C
  160 DO 163 I=1,2
  163 CALL RDISC(LW(I),IQU4,OM(1,I),IQNO-NPIECES)
C
      IF (ISW.NE.3) RETURN
C
      DO 168 I=3,4
  168 CALL RDISC(LW(I),IQU4,OM(1,I),IQNO-NPIECES)
C
      RETURN
C*****************************************************************
C*****************************************************************
      SUBROUTINE RASTER
C INTERNAL SUBROUTINE FOR PROGRAM BIVECO1 TO
C DETERMINE 1ST,LAST SUM FREQ'S FOR THIS RASTER
      GO TO (10,20,30,40,50,20,30,80,90,100,110,120,100,30,
     $20,20),ISQ
   10 IF (KPASS.EQ.1) GO TO 15
      IFIRST=3 ; ILAST=LPQU-1 ; RETURN
   15 IFIRST=2 ; ILAST=LPQU ; RETURN
   20 IFIRST=1 ; ILAST=LPATHE-1 ; RETURN
   30 IFIRST=1 ; ILAST=LPQU-1 ; RETURN
   40 IFIRST=LPATHE+2 ; ILAST=LP38   ; RETURN
   50 IF (KPASS.EQ.1) GO TO 55
      IFIRST=LPQU+3 ; ILAST=LPHALF-1 ; RETURN
   55 IFIRST=LPQU+2 ; ILAST=LPHALF ; RETURN
   80 IFIRST=LP38+2 ; ILAST=LPHALF ; RETURN
   90 IFIRST=LPQU+2 ; ILAST=LPHALF   ; RETURN
  100 IFIRST=LPATHE+1 ; ILAST=LP38-1 ; RETURN
  110 IFIRST=LPQU+1 ; ILAST=LPHALF-1 ; RETURN
  120 IFIRST=LP38+2 ; ILAST=LPHALF ; RETURN
C*****************************************************************
C*****************************************************************
      SUBROUTINE READW3
C INTERNAL SUBROUTINE FOR PROGRAM BIVECO1 TO FILL
C BUFFERS W31,W32 FROM CORE OR DISC APPROPRIATELY
      NRD=2*NPIECES
      GO TO (10,20,30,40,50,60,70,500,90,100,110,120,130,500,
     $60,60),ISQ
   10 IF (NCPT.LE.LPATHE) GO TO 20
      IF (ISW.EQ.1) KOL=3 ; MA=2 ; GO TO 300
      IF ((ISW.NE.2).OR.(NCPT.NE.LPATHE+1)) GO TO 15
```

```
C
      DO 12 I=1,2
   12 CALL RDISC(LW(I+4),IQU2,OM(1,I+2),IQNO)
C
   15 IF (ISW.NE.2) GO TO 18
      KOL=3 ; MA=2 ; GO TO 300
   18 GO TO 500
   20 CONTINUE
      IF (ISW.EQ.1) KOL=1 ; MA=1 ; GO TO 300
      IF ((ISW.EQ.2).OR.(LW31.EQ.LW21)) KOL=3 ; MA=1 ;GO TO 300
      KOL=1 ; MA=1 ;GO TO 300
   30 IF (ISW.EQ.2) GO TO 500
      IF (NCPT.GT.LPATHE) GO TO 35
      IF ((ISW.EQ.1).OR.(LW31.EQ.LW21)) KOL=3 ; MA=1 ;GO TO 300
      GO TO 500
   35 CONTINUE
      IF ((ISW.EQ.1).OR.(LW31.EQ.LW11)) KOL=1 ; MA=2 ;GO TO 300
      GO TO 500
   40 IF ((ISW.EQ.2).OR.(NCPT.GT.LPQU)) GO TO 500
      IF ((ISW.EQ.1).OR.(LW31.EQ.LW11)) KOL=1 ; MA=2 ;GO TO 300
      GO TO 500
   50 IF(ISW.EQ.3) GO TO 500
      IF (NCPT.NE.LP38+1) GO TO 58
C
      DO 51 I=1,ISOM
      W21(I)=0.
   51 W22(I)=0.
C
C
      DO 56 I=1,2
      GO TO (52,54),ISW
   52 CALL RDISC(LW(I),IQU4,OM(1,I+2),IQNO-NPIECES) ; GO TO 56
   54 CALL RDISC(LW(I+4),IQU4,OM(1,I+2),IQNO-NPIECES)
   56 CONTINUE
C
   58 CONTINUE
      IF (NCPT.LE.LP38) KOL=3 ; MA=3 ; GO TO 300
      KOL=3 ; MA=4 ; GO TO 300
   60 IF (ISW.EQ.3) GO TO 500
      KOL=3 ; MA=1 ; GO TO 300
   70 IF ((ISW.EQ.2).OR.(NCPT.LE.LPATHE)) GO TO 500
      IF ((ISW.EQ.1).OR.(LW31.EQ.LW21)) KOL=3 ; MA=2 ;GO TO 300
      GO TO 500
   90 IF ((ISW.EQ.2).OR.(NCPT.GT.LP38)) GO TO 500
      IF ((ISW.EQ.1).OR.(LW31.EQ.LW11)) KOL=1 ; MA=3;GO TO 300
      GO TO 500
  100 IF ((ISW.EQ.2).OR.(NCPT.LE.LPQU)) GO TO 500
      IF ((ISW.EQ.1).OR.(LW31.EQ.LW11)) KOL=1 ; MA=3 ;GO TO 300
      GO TO 500
  110 IF ((ISW.EQ.2).OR.(NCPT.LE.LP38)) GO TO 500
      IF ((ISW.EQ.1).OR.(LW31.EQ.LW11)) KOL=1 ; MA=4 ;GO TO 300
      GO TO 500
  120 CONTINUE
      IF ((ISW.EQ.1).OR.(LW31.EQ.LW11)) KOL=1 ; MA=4 ;GO TO 300
      GO TO 500
  130 IF ((ISW.EQ.2).OR.(NCPT.GT.LPQU)) GO TO 500
      IF ((ISW.EQ.1).OR.(LW31.EQ.LW21)) KOL=3 ; MA=2 ;GO TO 300
      GO TO 500
  300 LOC=1+NRD*(NCPT-(MA-1)*LPATHE-1)
C
```

```
C
      DO 350 I=1,NRD
      W31(I)=OM(LOC+I-1,KOL)
  350 W32(I)=OM(LOC+I-1,KOL+1)
C
C
      RETURN
  500 KREL=IDATSTART+NPIECES*(2*NCPT-1)
      CALL RDISC(LW31,KREL,W31,NRD)
      CALL RDISC(LW32,KREL,W32,NRD)
      RETURN
C*********************************************************************
C*********************************************************************
      SUBROUTINE FREQINIT
C INTERNAL SUBROUTINE FOR PROGRAM BIVEC01 TO INITIALIZE
C FREQUENCIES IF1,IF2 FOR A GIVEN SCAN.
      GO TO (10,20,30,40,10,60,70,80,90,100,110,120,130,140,
     $150,160),ISQ
   10 IF (KPASS.EQ.1) GO TO 15
      IF1=NCPT/2+1 ; IF2=NCPT/2
      IF (MOD(NCPT,2).EQ.0) IF2=NCPT/2-1
      RETURN
   15 IF1=NCPT/2 ; IF2=NCPT/2
      IF (MOD(NCPT,2).EQ.1) IF1=IF1+1
      RETURN
   20 IF2=-1
   25 IF1=-IF2+NCPT ; RETURN
   30 IF1C=LPATHE+1 ; IF2C=-1 ; IDIAG=LPATHE
   35 CONTINUE
      IF (NCPT.LE.IDIAG) IF1=IF1C ; IF2=-IF1+NCPT ; RETURN
      IF1=-IF2C+NCPT ; IF2=IF2C ; RETURN
   40 IF1C=LPATHE+1 ; IF2C=LPATHE   ; IDIAG=LPQU+1 ; GO TO 35
   60 IF2=-LPATHE-1 ; GO TO 25
   70 IF1C=LPQU+1 ; IF2C=-LPATHE-1 ; IDIAG=LPATHE ; GO TO 35
   80 IF1=LPQU+1
   85 IF2=-IF1+NCPT ; RETURN
   90 IF1C=LPQU+1 ; IF2C=LPATHE   ; IDIAG=LP38+1 ; GO TO 35
  100 IF1C=LPQU+1 ; IF2C=-1 ; IDIAG=LPQU ; GO TO 35
  110 IF1C=LP38+1 ; IF2C=-1 ; IDIAG=LP38 ; GO TO 35
  120 IF1=LP38+1 ; GO TO 85
  130 IF1C=LP38+1 ; IF2C=-LPATHE-1 ; IDIAG=LPQU ; GO TO 35
  140 IF1C=LP38+1 ; IF2C=-LPQU-1 ; IDIAG=LPATHE ; GO TO 35
  150 IF2=-LPQU-1 ; GO TO 25
  160 IF2=-LP38-1 ; GO TO 25
C*********************************************************************
C*********************************************************************
      SUBROUTINE BOUNDARY
C INTERNAL SUBROUTINE FOR PROGRAM BIVEC01
C TO TEST FOR END OF SCAN
      IBDY=0
      GO TO (10,20,30,40,50,60,70,80,90,100,110,120,130,140,
     $150,160),ISQ
   10 IDIAG=LPATHE ; IM1=LPATHE ; IM2=1 ; GO TO 500
   20 IM1=LPATHE ; GO TO 300
   30 IDIAG=LPATHE ; IM1=LPQU ; IM2=-LPATHE ; GO TO 500
   40 IDIAG=LPQU ; IM1=LPQU ; IM2=1 ; GO TO 500
   50 IDIAG=LP38 ; IM1=LPQU ; IM2=LPATHE+1 ; GO TO 500
   60 IM1=LPQU ; GO TO 300
   70 IDIAG=LPATHE ; IM1=LP38 ; IM2=-LPQU ; GO TO 500
   80 IDIAG=LPHALF ; IM2=LPATHE+1 ; GO TO 500
```

```
   90 IDIAG=LP38 ; IM1=LP38 ; IM2=1 ; GO TO 500
  100 IDIAG=LPQU ; IM1=LP38 ; IM2=-LPATHE ; GO TO 500
  110 IDIAG=LP38 ; IM1=LPHALF ; IM2=-LPATHE ; GO TO 500
  120 IDIAG=LPHALF ; IM2=1 ; GO TO 500
  130 IDIAG=LPQU ; IM1=LPHALF ; IM2=-LPQU ; GO TO 500
  140 IDIAG=LPATHE ; IM1=LPHALF ; IM2=-LP38 ; GO TO 500
  150 IM1=LP38 ; GO TO 300
  160 IM1=LPHALF ; GO TO 300
C
  300 IF (IF1.GT.IM1) IBDY=1
      RETURN
  500 IF (NCPT.GT.IDIAG) GO TO 550
      IF (IF2.LT.IM2) IBDY=1
      RETURN
  550 IF (IF1.GT.IM1) IBDY=1
      RETURN
C***************************************************************
C***************************************************************
      SUBROUTINE CBIN(BICOH,IPASS)
C INTERNAL SUBROUTINE FOR PROGRAM BISCAL03 TO DETERMINE
C CONFIDENCE LIMITS FOR BICOHERENCES
C HAS ENTRY POINTS CBIN TO PLACE GIVEN BICOHERENCE IN
C BIN AND CONFIDENCE TO INITIATE CONFIDENCE LIMIT CALCULATION
      IF (BICOH.LE.RBICLIM)  GO TO 100
      KOVER=KOVER+1
      IF (IPASS.EQ.1) WRITE (108,1000) IF1,IF2,IF3,BICOH
      IF (IPASS.EQ.2) WRITE (108,1000) -IF1,-IF2,-IF3,BICOH
 1000 FORMAT('ROT BICOH LIM EXCEEDED AT FREQ TRIPLET ',3(I3,X),
     $'  WITH VALUE ',G11.5)
      IF (IHI.NE.0) MTOT=MTOT+1
      RETURN
C
  100 CONTINUE
      NIB=INT(BICOH/BINSIZE)+1
      IF (NIB.GT.NBINS) GO TO 250
      KBIN(NIB)=KBIN(NIB)+1
      MTOT=MTOT+1
      RETURN
C
  250 CONTINUE
      IF (IPASS.EQ.1) WRITE (108,1025) BICOH,IF1,IF2,IF3
      IF (IPASS.EQ.2) WRITE (108,1025) BICOH,-IF1,-IF2,-IF3
 1025 FORMAT ('ROT BICOH OF ',F6.2,' AT FREQ TRIPLET ',3(I3,X),
     $' MET NEITHER BIN NOR LIM CRITERIA')
      MTOT=MTOT+1
      RETURN
C
C*************************************************************
      ENTRY CONFIDENCE
C**************************************************************
C OPTIONAL DISPLAY OF PARTITIONING OF BICOHERENCES
      IF (KB1.EQ.0) GO TO 320
      WRITE (108,1050)
 1050 FORMAT (/,T2,'BIN NO',T14,'BIN LIMITS',T34,
     $'NO OF ROT BIC''S',/)
C
C
      DO 300 I=1,NBINS
      WRITE (108,1100) I,(I-1)*BINSIZE,I*BINSIZE,KBIN(I)
 1100  FORMAT (T3,I3,T13,2(F4.2,4X),T34,I5)
```

```
  300 CONTINUE
C
C
  320 CONTINUE
C CONVERT NO OF ROT BIC'S IN KBIN TO TOTAL FRACTION
C AT INDEX VALUE
      KSUBTOT=0 ; I=0
      WRITE (108,1125) MTOT
 1125 FORMAT (/,'TOTAL NO OF ROT BIC''S SUBMITTED: ',I5,/)
      FTOT=FLOAT(MTOT)
C
  340 CONTINUE
      I=I+1
      IF (I.GT.NBINS) GO TO 350
C
      KSUBTOT=KSUBTOT+KBIN(I)
      ABIN(I)=FLOAT(KSUBTOT)
      ABIN(I)=ABIN(I)/FTOT
      GO TO 340
C
  350 CONTINUE
C OPTIONAL DISPLAY OF PARTIAL SUMS AT INDEX VALUE
      IF (KB2.EQ.0) GO TO 400
      WRITE (108,1150)
 1150 FORMAT (/,T32,'FRACT BELOW',/,T2,'BIN NO',T14,
     $'BIN LIMITS',T31,'UPPER BIN LIM',/)
C
C
      DO 380 I=1,NBINS
      WRITE (108,1200) I,(I-1)*BINSIZE,I*BINSIZE,ABIN(I)
 1200 FORMAT (T3,I3,T13,2(F4.2,4X),T34,F5.3)
  380 CONTINUE
C
C
  400 CONTINUE
C
C
C COMPUTE CONFIDENCE LIMITS
      I=0
C
  420 CONTINUE
      I=I+1
      IF (I.GT.NCONF) GO TO 600
      J=0
C
  450 CONTINUE
      J=J+1
      IF (J.GT.NBINS) WRITE (108,1250) CONF(I) ; BCL(I)=999. ;
     $GO TO 420
 1250 FORMAT ('BINS EXHAUSTED FOR CONFIDENCE LEVEL',F6.3)
      IF (ABIN(J).GE.CONF(I))  GO TO 470
      GO TO 450
C
  470 CONTINUE
C
      JH=J
      JL=J-1
      Y1=JL*BINSIZE
      Y2=JH*BINSIZE
      X1=0
```

```
      IF (JL.NE.0) X1=ABIN(JL)
      X2=ABIN(JH)
      X0=CONF(I)
      BCL(I)=Y1+(X0-X1)*(Y2-Y1)/(X2-X1)
      GO TO 420
  600 CONTINUE
C
C PRINT CONFIDENCE LEVELS
      WRITE (108,1400) NCONF,(CONF(I)*100.,BCL(I),I=1,NCONF)
 1400 FORMAT (/,T4,'CONFIDENCE',/,T6,'LEVEL',T21,'ROTARY ',
     $'BICOHERENCE',/,T5,'(PERCENT)',/,N(T7,F4.1,T24,F5.3,/),/)
      WRITE (108,1450) RBICLIM,KOVER
 1450 FORMAT ('NO. OF ROT BIC''S   GREATER THAN SPECIFIED ',
     $'MAXIMUM OF ',F5.2,' IS ',I5,/)
      RETURN
      END
```

- PROGRAM BPLOT -

```
C          *****  B P L O T  *****
C PROGRAM TO PLOT FILE OF BICOHERENCES WRITTEN BY
C  PROGRAM BISCAL
C PROGRAMMER:  GERARD H. MARTINEAU
C ORIGINATOR:  MELBOURNE G. BRISCOE
C       DATE:  JULY, 1977
      DIMENSION DATA(200,130),IBUF(1000),CONLV(15),
     $KARD(20),IDENT1(9),IDENT2(9),ICLK(4),LABL(33),CONFLV(15),
     $INAMES(9),IVAR(3)
      MD1=200 ; MD2=130
      NAMELIST ISTORE,NDECPL,CMPPT,IHISTORY,WIDTH
      ISTORE=NDECPL=1 ; CMPPT=-1. ; WIDTH=8.75
      IHISTORY=1
      INPUT
      READ (105,1111) KARD
 1111 FORMAT (20A4)
      WRITE (108,1122) KARD
 1122 FORMAT (X,20A4)
      DECODE (80,3333,KARD) NSERIES,LPIECE
 3333 FORMAT (2G)
      LPHALF=LPIECE/2 ; NFHALF=LPHALF/2
      IF (CMPPT.LT.0) CMPPT=WIDTH*2.54/LPHALF
      READ (105,1111) KARD
      WRITE (108,1122) KARD
      DECODE (80,2222,KARD) SAMPSEC,FREQTIC,NCONLV,
     $NCONLV,(CONLV(I),I=1,NCONLV),
     $NCONLV,(CONFLV(I),I=1,NCONLV)
 2222 FORMAT (3G,NG,NG)
      READ (105,1111) KARD
      WRITE (108,1122) KARD
      DECODE (80,4444,KARD) IDENT1,IDENT2
 4444 FORMAT (9A4,9A4)
C
C
C READ BICOHERENCES INTO ARRAY DATA(200,130)
C FIRST SET ARRAY DATA TO -999.
C
C
C     OUTPUT MD2,MD1,NFHALF,LPHALF
      DO 50 J=1,MD2
C
      DO 50 I=1,MD1
   50 DATA(I,J)=-999.
C
C     OUTPUT NSERIES,LPHALF,NFHALF,ISTORE
C
      IF (NSERIES.EQ.1) GO TO 60
      READ (ISTORE) ((DATA(I,J),I=LPHALF+2,LPHALF+NFHALF+1),
     $J=1,NFHALF)
C     OUTPUT 'B11'
      READ (ISTORE) ((DATA(I,J),I=NFHALF+1,LPHALF),J=1,NFHALF)
C     OUTPUT 'B22'
      READ (ISTORE) ((DATA(I,J),I=LPHALF+2,LPHALF+NFHALF+1),
     $J=NFHALF+1,LPHALF)
C     OUTPUT 'B33'
      READ (ISTORE) ((DATA(I,J),I=NFHALF+1,LPHALF),
     $J=NFHALF+1,LPHALF)
C     OUTPUT 'B44'
```

```
      READ (ISTORE) ((DATA(I,J),I=1,NFHALF),J=NFHALF+1,LPHALF)
C     OUTPUT 'B2'
      GO TO 80
   60 CONTINUE
C     OUTPUT 'B3'
      READ (ISTORE) ((DATA(I,J),I=1,NFHALF),J=1,NFHALF)
      READ (ISTORE) ((DATA(I,J),I=1,NFHALF),J=NFHALF+1,LPHALF)
   80 CONTINUE
      IF (IHISTORY.EQ.0) GO TO 85
C READ FOURIER LABELS
      READ (ISTORE) (INAMES(I),I=1,3),IVAR(1)
      READ (ISTORE) (INAMES(I),I=4,6),IVAR(2)
      READ (ISTORE) (INAMES(I),I=7,9),IVAR(3)
      READ (ISTORE) MF
      IF (MF.NE.21) GO TO 85
      READ (ISTORE) LABL
   85 CONTINUE
C     DO 90 K=1,11,10
C     WRITE (108,1111) ((DATA(I,J),J=K,K+9),I=1,30)
C1111 FORMAT (10(X,F6.2))
C  90 CONTINUE
C     OUTPUT 'B4'
      CALL PLOTS(IBUF,-1000)
C     OUTPUT 'B5'
      CALL PLOT(0.,0.5,-3)
C     OUTPUT 'B6'
      WRITE (108,1000) (CONLV(I),I=1,NCONLV)
 1000 FORMAT (/,'CONTOUR LEVELS:',/,(1H ,5G13.3,/))
      CALL DIMWH(DATA,MD1,MD2)
      CALL FLAGZ(-999.0)
      CALL NOLABL
      DEL=CMPPT/2.54
      AXLEN=LPHALF*DEL
      SAMPHR=SAMPSEC/3600.
      DFTOT=1./(2.*SAMPHR)
      DISTIC=AXLEN*FREQTIC/DFTOT
      AYLLEN=(INT(AXLEN/DISTIC)+1)*DISTIC
      AYHLEN=AXLEN/2.+0.25
      FIRSTV=-(AYLLEN/DISTIC)*FREQTIC
C     OUTPUT 'B7'
      IF (NSERIES.EQ.1) GO TO 120
C X-AXIS WITH TIC MARKS ONLY, STARTING AT ORIGIN
C     OUTPUT AXLEN,AYLLEN,AYHLEN,FREQTIC,DISTIC,NDECPL,FIRSTV
      CALL AXWJS(AXLEN,AXLEN,' ',-1,AXLEN,-90.,0.,1.,DISTIC,
     $-1,0,0,.12,.7,.0001,.14)
C X-AXIS STARTING ABOVE MAX Y VALUE
C     OUTPUT 'B71',AXLEN+AYHLEN,AXLEN,FREQTIC,DISTIC,NDECPL
      CALL AXDRAW(AXLEN+AYHLEN,AXLEN,'FREQ1 (CPH)',11,
     $AXLEN,-90.,0.,FREQTIC,DISTIC,NDECPL)
C X-AXIS STARTING BELOW MIN Y VALUE
C     OUTPUT 'B72',AXLEN-AYLLEN,AXLEN,FREQTIC,DISTIC,NDECPL
      CALL AXDRAW(AXLEN-AYLLEN,AXLEN,'FREQ1 (CPH)',-11,
     $AXLEN,-90.,0.,FREQTIC,DISTIC,NDECPL)
C Y-AXIS FOR TWO SERIES
C     OUTPUT 'B73',AXLEN-AYLLEN,AXLEN,AYLLEN+AYHLEN,FIRSTV,DISTIC
      CALL AXDRAW(AXLEN-AYLLEN,AXLEN,'FREQ2 (CPH)',11,
     $AYLLEN+AYHLEN,0.,FIRSTV,FREQTIC,DISTIC,NDECPL)
      GO TO 150
  120 CONTINUE
C X-AXIS FOR ONE SERIES
```

```
      CALL AXDRAW(0.,AXLEN,'FREQ1 (CPH)',-11,AXLEN,-90.,0.,
     $FREQTIC,DISTIC,NDECPL)
C Y-AXIS FOR ONE SERIES
      CALL AXDRAW(0.,AXLEN,'FREQ2 (CPH)',11,AYHLEN,0.,0.,
     $FREQTIC,DISTIC,NDECPL)
  150 CONTINUE
C PLOT IDENTIFICATION
      IF (NSERIES.EQ.1) GO TO 160
      EXPLOT=AXLEN+AYHLEN
      GO TO 165
  160 EXPLOT=AYHLEN
  165 WYEPLOT=AXLEN
      IF (NSERIES.EQ.1) GO TO 170
      CALL SYMBOL(EXPLOT+7.0,WYEPLOT-1.8,0.315,
     $'CROSS BICOHERENCE',-90.,17)
      GO TO 175
  170 CALL SYMBOL(EXPLOT+7.0,WYEPLOT-2.0,0.315,
     $'AUTO BICOHERENCE',-90.,16)
  175 CONTINUE
      IF (IHISTORY.EQ.0) GO TO 179
      CALL SYMBOL(EXPLOT+6.5,WYEPLOT-1.6,0.175,
     $'ORIGINAL FILE NAMES:',-90.,20)
      CALL SYMBOL(EXPLOT+6.5,WYEPLOT-5.7,0.175,
     $'VARIABLE NUMBERS:',-90.,17)
      CALL SYMBOL(EXPLOT+6.2,WYEPLOT-0.5,0.14,'FREQUENCY 1:',
     $-90.,12)
      CALL SYMBOL(EXPLOT+5.9,WYEPLOT-0.5,0.14,'FREQUENCY 2:',
     $-90.,12)
      CALL SYMBOL(EXPLOT+5.6,WYEPLOT-0.5,0.14,'FREQUENCY 3:',
     $-90.,12)
      IF (MF.NE.21) GO TO 177
      CALL SYMBOL(EXPLOT+5.1,WYEPLOT-2.9,0.175,
     $'PROCESSING HISTORY',-90.,18)
  177 CONTINUE
      CALL SYMBOL(EXPLOT+6.2,WYEPLOT-2.5,0.14,INAMES,-90.,12)
      CALL SYMBOL(EXPLOT+5.9,WYEPLOT-2.5,.14,INAMES(4),-90.,12)
      CALL SYMBOL(EXPLOT+5.6,WYEPLOT-2.5,.14,INAMES(7),-90.,12)
      FVAR=FLOAT(IVAR(1))
      CALL NUMBER(EXPLOT+6.2,WYEPLOT-7.1,0.14,FVAR,-90.,-1)
      FVAR=FLOAT(IVAR(2))
      CALL NUMBER(EXPLOT+5.9,WYEPLOT-7.1,0.14,FVAR,-90.,-1)
      FVAR=FLOAT(IVAR(3))
      CALL NUMBER(EXPLOT+5.6,WYEPLOT-7.1,0.14,FVAR,-90.,-1)
      IF (MF.NE.21) GO TO 179
      CALL SYMBOL(EXPLOT+4.8,WYEPLOT-0.7,0.105,LABL,-90.,72)
      CALL SYMBOL(EXPLOT+4.5,WYEPLOT-1.3,.105,LABL(19),-90.,60)
  179 CONTINUE
      CALL SYMBOL(EXPLOT+1.15,WYEPLOT-1.0,0.210,IDENT1,-90.,36)
      CALL SYMBOL(EXPLOT+0.85,WYEPLOT-1.0,0.210,IDENT2,-90.,36)
      CALL TODAY(ICLK)
      CALL SYMBOL(EXPLOT+3.0,WYEPLOT-0.6,0.175,'TIME OF PLOT:',
     $-90.,13)
      CALL SYMBOL(EXPLOT+2.7,WYEPLOT-0.3,0.175,ICLK,-90.,16)
      CALL SYMBOL(EXPLOT+3.8,WYEPLOT-3.8,0.140,
     $'CONTOUR LEVELS AND PERCENT CONFIDENCE',-90.,37)
      BIASX=BIASY=0.
C
C
      IF (NCONLV.GT.10) NCONLV=10
      DO 185 I=1,NCONLV
```

```
      IF (I.EQ.6)                    BIASX=0. ; BIASY=BIASY-3.0
      CALL NUMBER(EXPLOT+3.3+BIASX,WYEPLOT-4.0+BIASY,0.210,
     $CONLV(I),-90.,3)
      CALL NUMBER(EXPLOT+3.3+BIASX,WYEPLOT-5.4+BIASY,0.210,
     $CONFLV(I),-90.,1)
      BIASX=BIASX-0.4
  185 CONTINUE
C
C
C DRAW BOUNDARIES
      CALL PLOT(0.,0.,3)
C     OUTPUT 'B8'
      IF (NSERIES.EQ.1) GO TO 200
      CALL PLOT(AXLEN,AXLEN,2)
      CALL PLOT(1.5*AXLEN,AXLEN/2.,2)
      CALL PLOT(AXLEN,0.,2)
      CALL PLOT(0.,0.,2)
      GO TO 205
  200 CONTINUE
      CALL PLOT(0.,AXLEN,3)
      CALL PLOT(AXLEN/2.,AXLEN/2.,2)
      CALL PLOT(0.,0.,2)
  205 CONTINUE
C CALLS TO SYMBOL AND NUMBER
      GO TO (210,220),NSERIES
  210 CALL GRID(DEL,AXLEN-DEL,DEL,-DEL)
C     OUTPUT DEL,NFHALF,LPHALF,'ONE'
      CALL WHCNTR(DATA,1,NFHALF,1,LPHALF,NCONLV,CONLV)
      CALL PLOT(0.,0.,3)
      CALL PLOT(1.5*AXLEN,-0.5,999)
      GO TO 250
  220 CALL GRID(0.,AXLEN-DEL,DEL,-DEL)
C     OUTPUT DEL,NFHALF,LPHALF,'TWO'
      CALL WHCNTR(DATA,1,LPHALF+NFHALF+1,1,LPHALF,NCONLV,CONLV)
      CALL PLOT(0.,0.,3)
      CALL PLOT(2.5*AXLEN,-0.5,999)
  250 CONTINUE
      STOP 'NORMAL PROGRAM COMPLETION'
      END
```

- PROGRAM FOURIER -

```
C      <<< FOURIER >>>
C PROGRAM TO CREATE RWDISC FILE OF FOURIER COEFFICIENTS
C FROM RWDISC FILE OF SERIES
C INPUT FILE HAS 21 LOGICAL FILES, THE LAST BEING
C IMAGE OF ORIGINAL TAPDIS FILE.
C PROGRAMMER: G. MARTINEAU VERSION 02 1/12/78
       COMMON WK(33000),
      $AMEAN(150),MPT(150),KPT(150),NSSMP(2),
      $ICLK(4),LABL(33),IOVERLAP,IHANN,
      $ND02,NPIN,NDIM,IPREW,ISUBSAMP,NPIECES,LPIECE,
      $R2,ERRY,ERRA0,ERRA1,ISIGN,ITAPER,MFFT,
      $NFILES,ISUBT,MF,IFIRST,ILAST
       DIMENSION X(450),Y(450)
       EQUIVALENCE (NSSMP(1),NSUBSAMP)
       CALL CTRL
X      OUTPUT 'F1'
       CALL DPAR(NERR,MF,1,1,KSUM,512)
C
C
       DO 500 LF=IFIRST,ILAST
X      OUTPUT 'F2'
       CALL INPUT(LF)
X      OUTPUT 'F3'
       CALL KOEFF(X,Y,LPIECE)
X      OUTPUT 'F4'
       CALL OUTOMEGA(LF)
X      OUTPUT 'F5'
   500 CONTINUE
C
C
       CALL TODAY(ICLK)
       CALL LABEL
X      OUTPUT 'F6'
       CALL CDISC
       STOP 'PROCESSING COMPLETED'
C
C
       SUBROUTINE LABEL
C INTERNAL SUBROUTINE FOR PROGRAM FOURIER TO CREATE
C LABEL AND OUTPUT TO LOG. FILE 21 AND WRITE
C ON LINE PRINTER
       NDATL=NPIECES*LPIECE
       NFILES=ILAST-IFIRST+1
       ENCODE (132,1000,LABL,NOCH) NFILES,NPIN,IPREW,ISUBSAMP,
      $NSUBSAMP,LPIECE,NPIECES,IOVERLAP,ISUBT,IHANN,ICLK
  1000 FORMAT ('NO LF''S ',I2,'*NO WDS DATA ',I5,'*PREW ',I1,
      $'*SUBSAMP ',I1,'*NSUBSAMP ',I3,'*PC SIZE ',I5,'*NO PCS ',
      $I3,'*OLAP ',I1,'*ISUBT ',I1,'*HANN ',I1,' *CREATED ',4A4)
       IF (MF.NE.21) GO TO 150
C WRITE L.F. 21 ON DISC
       CALL WDISC(21,1,LABL,33)
   150 CONTINUE
C WRITE LABEL TO LINE PRINTER
       WRITE(108,2000) LABL
  2000 FORMAT(/,'LABEL AS WRITTEN ON L.F. 21 IF MF=21:',
      $/,1H ,33A4,/)
       RETURN
       END
```

```
C*********************************************************************
C*********************************************************************
      SUBROUTINE LINREG(NPTS,X,Y,A0,A1,R2,ERRY,ERRA0,ERRA1)
C     *************************************************************
C
C     LINEAR REGRESSION OF Y ON X
C
      DOUBLE PRECISION SX,SY,SX2,SY2,SXY,SXX,SYY,A0,A1
      DIMENSION X(1),Y(1),TEMP(450)
C     CALCULATE SUMS OF SQUARES, ETC.
      CALL SUMM(NPTS,X,SX)
      CALL SUMM(NPTS,Y,SY)
      DO 10 I=1,NPTS
   10 TEMP(I)=X(I)*X(I)
      CALL SUMM(NPTS,TEMP,SX2)
      DO 20 I=1,NPTS
   20 TEMP(I)=Y(I)*Y(I)
      CALL SUMM(NPTS,TEMP,SY2)
      DO 30 I=1,NPTS
   30 TEMP(I)=X(I)*Y(I)
      CALL SUMM(NPTS,TEMP,SXY)
      SXX=SX2-SX*SX/NPTS
      SYY=SY2-SY*SY/NPTS
      SSXY=SXY-SX*SY/NPTS
C     CALCULATE REGRESSION LINE Y=A0+A1*X
      A1=SSXY/SXX
      A0=SY/NPTS-A1*SX/NPTS
C     CALCULATE CORRELATION COEFFICIENT
      R2=SSXY*SSXY/SXX/SYY
C      CALCULATE STANDARD ERRORS OF ESTIMATE
      ERRY=DSQRT((SY2-A0*SY-A1*SXY)/(NPTS-2))
      ERRA0=ERRY*DSQRT(SX2/NPTS/SXX)
      ERRA1=ERRY/DSQRT(SXX)
      RETURN
      END
C*********************************************************************
C*********************************************************************
      SUBROUTINE TRENDR(NPTS,X,A0,A1)
C     ********************************
C
C     REMOVES LINEAR REGRESSION LINE FOUND BY LINREG
C
      DOUBLE PRECISION A0,A1
      DIMENSION X(1)
      DO 10 I=1,NPTS
   10 X(I)=X(I)-A0-A1*I
      RETURN
      END
C*********************************************************************
C*********************************************************************
      SUBROUTINE SUMM(N,X,SUMX)
C     *************************
C
      DOUBLE PRECISION SUM1,SUMX
      DIMENSION X(1)
      SUM1=0
      DO 10 I=1,N
   10 SUM1=SUM1+X(I)
      SUMX=SUM1
      RETURN
```

```
      END
C***********************************************************************
C***********************************************************************
      SUBROUTINE CTRL
C FOR PROGRAM FOURIER, TO INITIALIZE,
C AND READ IN CONTROL PARAMETERS
      COMMON WK(33000),
     $AMEAN(150),MPT(150),KPT(150),NSSMP(2),
     $ICLK(4),LABL(33),IOVERLAP,IHANN,
     $NDO2,NPIN,NDIM,IPREW,ISUBSAMP,NPIECES,LPIECE,
     $R2,ERRY,ERRAO,ERRAL,ISIGN,ITAPER,MFFT,
     $NFILES,ISUBT,MF,IFIRST,ILAST
      DIMENSION KFFT(84)
      DATA KFFT/
     $   12,  16,  20,  24,  32,  36,  40,  48,  60,  64,
     $   72,  80,  96, 100, 108, 120, 128, 144, 160, 180,
     $  192, 200, 216, 240, 256, 288, 300, 320, 324, 360,
     $  384, 400, 432, 480, 500, 512, 540, 576, 600, 640,
     $  648, 720, 768, 800, 864, 900, 960, 972,1000,1024,
     $ 1080,1152,1200,1280,1296,1440,1500,1536,1600,1620,
     $ 1728,1800,1920,1944,2000,2048,2160,2304,2400,2500,
     $ 2560,2592,2700,2880,2916,3000,3072,3200,3240,3456,
     $ 3600,3840,3888,4000/
      EQUIVALENCE (NSSMP(1),NSUBSAMP)
      IPREW=ISUBSAMP=NSUBSAMP=ISUBT=0
      MF=21
      IOVERLAP=IHANN=1
      NAMELIST MF
      READ(105,1000) IFIRST,ILAST,NDO2,IPREW,ISUBSAMP,
     $ISUBSAMP,(NSSMP(I), I=1,ISUBSAMP),LPIECE,IOVERLAP,ISUBT,
     $IHANN
 1000 FORMAT (5G,N(G),4G)
      DO 90  I=1,84
      IF (LPIECE.EQ.KFFT(I)) GO TO 200
   90 CONTINUE
      WRITE (108,2000) LPIECE,KFFT ; STOP
 2000 FORMAT ('PIECE LENGTH OF ',I4,' INVALID.MUST BE ',
     $'FROM THE FOLLOWING SET:',/,(X,25I5))
  200 CONTINUE
      NPIN=NDO2
      IF (ISUBSAMP.EQ.0) GO TO 100
C NSUBSAMP MUST BE ODD FOR TRIANGULAR WEIGHTING
      IF(MOD(NSUBSAMP,2) .EQ. 0) WRITE(108,1111);
     $OUTPUT NSUBSAMP;STOP
 1111 FORMAT('SUBSAMPLE LENGTH MUST BE ODD')
      NDO2 = NDO2/NSUBSAMP
  100 CONTINUE
      INPUT
      IF ((IHANN.EQ.0).AND.(IOVERLAP.EQ.1)) IOVERLAP=0 ;
     $OUTPUT 'OVERLAP POSSIBLE ONLY IF HANN. IOVERLAP SET TO 0'
      IF ((IOVERLAP.EQ.0).AND.((LPIECE.GT.450).OR.
     $(LPIECE.LT.NDO2/150))) OUTPUT LPIECE,
     $'PIECE LENGTH DOES NOT SATISFY RESTRICTION FOR NO OVERLAP
     $'; STOP
      IF ((IOVERLAP.NE.0).AND.((LPIECE.GT.450).OR.
     $(LPIECE.LT.NDO2/75))) OUTPUT LPIECE,
     $'PIECE LENGTH DOES NOT SATISFY RESTRICTION FOR OVERLAP';
     $STOP
      IF (MF.GT.21) OUTPUT 'MF CANNOT EXCEED 21' ; STOP
      IF (((MF.LT.21).AND.(ILAST.GT.MF)).OR.
```

```
     $((MF.EQ.21).AND.(ILAST.GT.20))) WRITE (108,2222)
2222 FORMAT ('LAST LOG. FILE CANNOT EXCEED MF OR 20')
      RETURN
      END
C****************************************************************
C******************************************************************
      SUBROUTINE INPUT(LF)
C FOR PROGRAM FOURIER, TO INPUT AND PREPARE
C AN RWDISC LOG. FILE FOR THE TRANSFORM.
C DOES PREWHITENING,SAMPLING AND SPLITTING
C INTO PIECES
      COMMON WK(33000),
     $AMEAN(150),MPT(150),KPT(150),NSSMP(2),
     $ICLK(4),LABL(33),IOVERLAP,IHANN,
     $NDO2,NPIN,NDIM,IPREW,ISUBSAMP,NPIECES,LPIECE,
     $R2,ERRY,ERRA0,ERRA1,ISIGN,ITAPER,MFFT,
     $NFILES,ISUBT,MF,IFIRST,ILAST
      EQUIVALENCE (NSSMP(1),NSUBSAMP)
X     OUTPUT 'I1'
C READ IN DATA FROM RWDISC FILE, STARTING AFTER LABELS
      CALL RDISC(LF,5,WK,NPIN)
      IF(IPREW .EQ. 1) CALL PREWHITE
      IF(ISUBSAMP .EQ. 0) GO TO 150
C
C
C DO TRIANGULAR WEIGHTING
C
      DO 100 I=1,NDO2
      I1 = 1+(I-1)*NSUBSAMP
      MT = NSUBSAMP/2+1
      ASUM = 0.
C
      DO 80 J=1,NSUBSAMP
   80 ASUM = ASUM+((MT-ABS(MT-J))/(MT*MT))*WK(I1+J-1)
C
      WK(I) = ASUM
  100 CONTINUE
C
C
  150 CONTINUE
X     OUTPUT 'I2'
C
      IF (IOVERLAP.EQ.0) GO TO 300
C
      NDIM = 2*NDO2
      NPIECES = 2*NDO2/LPIECE-1
C CALCULATE LENGTH OF TOTAL PIECE TO BE MOVED
      NMOVE = NPIECES/2
      LPHALF = LPIECE/2
      IF(NMOVE .EQ. 0) MPT(1) = 1; RETURN
      NMLP = NMOVE*LPIECE
C FILL ARRAY OF INDICES
C
C
      DO 200 I=1,NPIECES
      MPT(I) = (1+(I/2)*LPIECE)*MOD(I,2)+
     $(1+(I/2-1)*LPIECE+NDO2)*MOD(I+1,2)
  200 CONTINUE
C
C
```

```
C ORGANIZE ARRAY WK
C
C
X      OUTPUT 'I3'
       DO 250 I=1,NMLP
       WK(NDO2+I) = WK(LPHALF +I)
   250 CONTINUE
       GO TO 800
C
C NO SPLITTING INTO PIECES
   300 CONTINUE
       NDIM=NDO2
       NPIECES=NDO2/LPIECE
C FILL ARRAY OF INDICES
C
C
       DO 400 I=1,NPIECES
       MPT(I)=1+(I-1)*LPIECE
   400 CONTINUE
C
C
   800 CONTINUE
C
C
       RETURN
       SUBROUTINE PREWHITE
C INTERNAL SUBROUTINE FOR SUBROUTINE INPUT
C TO PREWHITEN DATA SERIES
C
C
X      OUTPUT 'I4'
       DO 100 I=1,NDO2-1
   100 WK(I) = WK(I+1)-WK(I)
C
C
       WK(NDO2) = WK(NDO2-1)
       RETURN
       END
C*****************************************************************************
C*****************************************************************************
       SUBROUTINE  KOEFF(X,Y,LPIECE)
C FOR PROGRAM FOURIER, TO DO FFT ON EACH PIECE
C AND REPLACE N POINTS OF PIECE BY
C COEFFICIENTS A0,A1,.....A(N/2),B1,B2,...B(N/2-1)
C ALSO SUBTRACTS MEAN OR LINEAR TREND, AND FILTERS
       COMMON WK(33000),
     $AMEAN(150),MPT(150),KPT(150),NSSMP(2),
     $ICLK(4),LABL(33),IOVERLAP,IHANN,
     $NDO2,NPIN,NDIM,IPREW,ISUBSAMP,NPIECES,LPCDUM,
     $R2,ERRY,ERRA0,ERRA1,ISIGN,ITAPER,MFFT,
     $NFILES,ISUBT,MF,IFIRST,ILAST
X      OUTPUT NDO2,NDIM,IPREW,ISUBSAMP,NPIECES,NFILES,ISUBT
X      OUTPUT MFFT,ISIGN,ITAPER
       EQUIVALENCE (NSSMP(1),NSUBSAMP)
       DIMENSION X(LPIECE),Y(LPIECE)
       DOUBLE PRECISION A0,A1
X      OUTPUT LPIECE
       LPHALF = LPIECE/2
C
C
```

```
      DO 800 I=1,NPIECES
      M=MPT(I)
      AMEAN(I) = 0.
C PULL PIECE OUT  OF WORKING ARRAY
C
      DO 40 J=1,LPIECE
      X(J)  = WK(M+J-1)
   40 Y(J)  = J
X     WRITE (108,1111) I,MPT(I),(X(J),J=1,LPIECE)
X1111 FORMAT ('DATA PIECE',2X,'PIECE NO. ',I2,10X,'MPT(I)=',
X    $I5,/,6(1H ,5G13.5,/),(1H ,2G13.5,//))
C
      DO 80 J=1,LPIECE
   80 AMEAN(I) = AMEAN(I) + X(J)
      AMEAN(I) = AMEAN(I)/LPIECE
      GO TO  (400,200,100,85),ISUBT  + 1
   85 CONTINUE
C SUBTRACT   MEAN OF EACH PIECE
C
      DO 90 J=1,LPIECE
   90 X(J)  = X(J)  - AMEAN(I)
C
X     OUTPUT 'K2'
      GO TO  400
  100 CONTINUE
C CALCULATE  TREND  BY  LINEAR REGRESSION
C AND REMOVE IT
      CALL LINREG(LPIECE,Y,X,A0,A1,R2,ERRY,ERRA0,ERRA1)
      CALL TRENDR(LPIECE,X,A0,A1)
      GO TO  400
  200 CONTINUE
C TREND  REMOVAL BASED ON ENDPOINTS
X     OUTPUT 'K3'
      AA = (X(LPIECE)-X(1))/(LPIECE-1)
      BB = (LPIECE*X(1)-X(LPIECE))/(LPIECE-1)
C
      DO 300 J=1,LPIECE
      X(J)  = X(J)-AA*J-BB
  300 CONTINUE
C
  400 CONTINUE
C DO FOURIER TRANSFORM
X     OUTPUT 'K4',LPIECE
      INDCTR=1
      DO 450 J=1,LPIECE
  450 Y(J)=0.
      CALL HARM1(X,Y,INDCTR,LPIECE)
C
      IF (IHANN.EQ.0) GO TO 685
      C=SQRT(8./3.)
C
      DO 500 K=1,LPIECE
  500 X(K)=C*X(K)
C
  550 CONTINUE
      SAV1=X(1)
C
      DO 650 K=1,LPHALF-1
  650 X(K)=0.5*X(K+1)-0.25*(X(K)+X(K+2))
C
```

```
      DO 655 K=1,LPHALF-1
  655 X(1+LPHALF-K)=X(LPHALF-K)
C
      X(1)=SAV1
      SAV1=X(LPHALF+2)
C
      DO 670 K=LPHALF+2,LPIECE-2
  670 X(K)=0.5*X(K+1)-0.25*(X(K)+X(K+2))
C
      DO 675 K=1,LPHALF-3
  675 X(LPIECE-K)=X(LPIECE-K-1)
C
      X(LPHALF+2)=SAV1
  685 CONTINUE
C RESTORE  PIECE TO  WORKING ARRAY
C
X     OUTPUT M
      DO 700 J=1,LPIECE
  700 WK(M+J-1) = X(J)
C
X     WRITE (108,2222) I,MPT(I),(X(J),J=1,LPIECE)
X2222 FORMAT ('COEFFICIENTS',2X,'PIECE NO. ',I2,10X,'MPT(I)=',
X    $I5,/,6(1H ,5G13.5,/),(1H ,2G13.5,//))
  800 CONTINUE
C
C
      RETURN
      END
C*****************************************************************
C*****************************************************************
      SUBROUTINE OUTOMEGA(LF)
C FOR PROGRAM FOURIER TO OUTPUT COEFFICIENTS TO RWDISC
C FILE IN ORDER OF INCREASING FREQUENCY AND
C ORDER A,B AT A GIVEN FREQUENCY, I.E.:
C A(0),A(1),B(1),...,A(N/2-1),B(N/2-1),A(N/2)
      COMMON WK(33000),
     $AMEAN(150),MPT(150),KPT(150),NSSMP(2),
     $ICLK(4),LABL(33),IOVERLAP,IHANN,
     $NDO2,NPIN,NDIM,IPREW,ISUBSAMP,NPIECES,LPIECE,
     $R2,ERRY,ERRAO,ERRA1,ISIGN,ITAPER,MFFT,
     $NFILES,ISUBT,MF,IFIRST,ILAST
      EQUIVALENCE (NSSMP(1),NSUBSAMP)
      DIMENSION BUF(2048)
      KOEFF1=1;LPHALF=LPIECE/2;KA=0;KB=LPHALF+1;IAB=-1;M=1
      IF (NPIECES.EQ.1) GO TO 200
      NCOEFF=2048/NPIECES
      KOEFF2=NCOEFF
      IF (NCOEFF.GE.LPIECE) KOEFF2=LPIECE
C FILL BUF FROM KOEFF1 THRU KOEFF2
   50 CONTINUE
      M=1
C
C
      DO 100 I=KOEFF1,KOEFF2
      IAB=-IAB
      IF (I.EQ.2) IAB=-IAB
      IF (IAB.EQ.1) KA=KA+1 ; KUSE=KA
      IF (IAB.EQ.-1) KB=KB+1 ; KUSE=KB
X     IF (KUSE.GT.LPIECE) STOP 'CHK LOGIC 0000'
C
```

```
          DO 80 J=1,NPIECES
          LOC=MPT(J)-1
          BUF(M)=WK(LOC+KUSE)
          M=M+1
       80 CONTINUE
C
      100 CONTINUE
C
C
C WRITE TO DISC
          IWR=(KOEFF1-1)*NPIECES+1
          NWR=(KOEFF2-KOEFF1+1)*NPIECES
          CALL WDISC(LF,4+IWR,BUF,NWR)
C RETURN IF HAVE WRITTEN LPIECE COEFFICIENTS
          IF (KOEFF2.EQ.LPIECE) RETURN
X         IF (KOEFF2.GT.LPIECE) STOP 'CHK LOGIC 1111'
          KOEFF1=KOEFF2+1
          KOEFF2=KOEFF2+NCOEFF
          IF (KOEFF2.GE.LPIECE) KOEFF2=LPIECE
          GO TO 50
X 150     STOP 'CHK LOGIC 2222'
  200     RETURN
          END
```

- PROGRAM FRAGTAP -

```
C            <<<<<<<  F R A G T A P  >>>>>>>
C PROGRAM TO READ TAPDIS FILES AND CREATE CONSECUTIVE FILES
C FOR LATER TRANSMITTAL TO NON-TAPDIS RWDISC FILE
C PROGRAMMER:  G. MARTINEAU  VERSION 01  3/9/77
      DIMENSION FILE(18000),IFILE(2500),JDCB(25),NSEQ(25),
     $NV(25),ILF(25,4),JFILES(25),NCYC(25),IFILNAM(3),
     $ISVAR(4),ILAB(9)
      EQUIVALENCE (FILE,IFILE),(ILAB,IFILNAM),(ILAB(4),INVAR),
     $(ILAB(5),ICYCLES),(ILAB(6),ISVAR)
      MFILE=7
      CALL DPAR(NERR,MFILE,1,1,KSUM)
      WRITE (108,1000)
 1000 FORMAT ('INPUT RECORDS FOLLOW (IN GENERALIZED FORMAT)')
      READ (105,1500) NF,NF,(JDCB(I),I=1,NF),LDCB
 1500 FORMAT (G,NG,G)
      WRITE (108,1500) NF,NF,(JDCB(I),I=1,NF),LDCB
C
C
      DO 100 I=1,NF
      READ (105,2000) NSEQ(I),NV(I),NV(I),(ILF(I,K),K=1,NV(I))
 2000 FORMAT (2G,NG)
      WRITE (108,2200) NSEQ(I),NV(I),NV(I),(ILF(I,K),K=1,NV(I))
 2200 FORMAT(1H ,2G,NG)
  100 CONTINUE
C
C
C CALCULATE STARTING LOCATIONS OF BLOCKS IN TAPDIS FILE
      JFILES(1)=1
      CALL RDISC(1,100,NOFILES,1)
X     OUTPUT NOFILES
      IF (NERR.NE.0) OUTPUT NERR ; STOP 2
      LENGTH=100*NOFILES
      CALL RDISC(1,1,IFILE,LENGTH)
X     OUTPUT LENGTH
      IF (NERR.NE.0) OUTPUT NERR ; STOP 3
C
C
      DO 180 I=1,NOFILES
  180 NCYC(I)=IFILE(100*(I-1)+9)
C
      DO 200 I=2,NOFILES
  200 JFILES(I)=JFILES(I-1)+IFILE(100*(I-2)+9)
C
C
      WRITE (108,2500) (I,JFILES(I),I=1,NOFILES)
 2500 FORMAT (/,T4,'SEQUENTIAL',T21,'REL LOCATION',/,
     $T2,'INPUT FILE READ',T22,'IN TAPDIS',/,T5,'BY TAPDIS',
     $T21,'FILE (KREL)',/,(T8,I2,T23,I6))
      WRITE (108,2700)
 2700 FORMAT (/,T5,'OUTPUT FILE SUMMARY IN ORDER OF ACCESS',//,
     $T2,'OUTPUT',T29,'BUOY TAPE VAR''S',T50,'TOTAL WORDS',/,
     $T3,'DCB',T13,'TAPE FILE',T28,'ACCESSED THIS FILE',T50,
     $'IN OUTPUT FILE',/)
C
C
      DO 240 I=1,NF
      IXDCB=JDCB(I)
      KALL=NSEQ(I)
```

```
            ICYCLES=NCYC(KALL)
            INVAR=NV(I)
C
            DO 230 N=1,3
        230 IFILNAM(N)=IFILE(100*(KALL-1)+N)
C
            DO 235 N=1,INVAR
        235 ISVAR(N)=IFILE(100*(KALL-1)+4+N)
C
            WRITE (108,3000) IXDCB,IFILNAM,INVAR,(ISVAR(N),N=1,INVAR)
           $,ICYCLES+9
       3000 FORMAT (T2,I3,T12,3A4,T29,N(I2,2X),T52,I5)
X           OUTPUT I,IXDCB
            CALL BUFFER OUT(IXDCB,1,ILAB,9,ISTAT)
            IF (ISTAT.NE.2) OUTPUT ISTAT; STOP 3
        240 CONTINUE
C
C
C WRITE LABELS TO BINARY SEQ. FILE
            CALL BUFFER OUT(LDCB,1,IFILE,LENGTH,ISTAT)
X           OUTPUT 'LABELS OUTPUT'
            IF (ISTAT.NE.2) OUTPUT ISTAT ; STOP 1
C READ AND WRITE TAPDIS FILES
C
C
            DO 300 I=1,NF
            IXDCB=JDCB(I)
            KALL=NSEQ(I)
            ICYCLES=NCYC(KALL)
            KSTART=JFILES(KALL)
C
            DO 250 J=1,NV(I)
            IVAR=ILF(I,J)
X           OUTPUT IVAR,KSTART,ICYCLES
            CALL RDISC(IVAR,KSTART,FILE,ICYCLES)
            IF (NERR.NE.0) OUTPUT NERR ; STOP 4
X           OUTPUT I,IXDCB,ICYCLES
            CALL BUFFER OUT(IXDCB,1,FILE,ICYCLES,ISTAT)
            IF (ISTAT.NE.2) OUTPUT ISTAT ; STOP 4
        250 CONTINUE
C
        300 CONTINUE
C
C
            STOP 'OUTPUT COMPLETED'
            END
```

- PROGRAM GENRAN -

```
C              <<<<  G E N R A N O 4  >>>>
C PROGRAM TO GENERATE RWDISC FILE OF KNOWN SIGNALS.
C USES IPC PROGRAM NORAN FOR PSEUDO-RANDOM NOISE.
C PROGRAMMER: G. MARTINEAU    8/2/77
       DOUBLE PRECISION T,PI,TWOPI,DFLS,DFLTH,FL,DFMS,DFMTH
       DOUBLE PRECISION
      $W11(20),W12(20),W21(20),W22(20),W31(20),W32(20),
      $W41(20),W42(20),
      $T11(20),T12(20),T21(20),T22(20),T31(20),T32(20),
      $T41(20),T42(20),
      $AH1(20),AH2(20),AH3(20),AH4(20),AH5(20),AH6(20),
      $TH1(20),TH2(20),TH3(20),TH4(20),TH5(20),TH6(20)
       DIMENSION KHARM(20),XS1(20),XS2(20),XS3(20),XS4(20),
      $BMEAN(20),SDE(20),IXA(20)
       DIMENSION LABL(4)
       COMMON B1(2560),KG(20),KNG(20),XG(20),XNG(2,20),FUND(20),
      $AH1,AH2,AH3,AH4,AH5,AH6,
      $TH1,TH2,TH3,TH4,TH5,TH6,
      $IDIM,IFL,TWOPI,LENGTH
       NAMELIST IG,ING,AMEAN,STDEV,AG,ANG,AS1,AS2,AS3,AS4,IHARM,
      $F11,F12,F21,F22,F31,F32,P11,P12,P21,P22,P31,P32,KSAME,IX,
      $F41,F42,P41,P42,
      $ASQ,ACUBE,HA1,HA2,HA3,HA4,HA5,HA6,FF,
      $HP1,HP2,HP3,HP4,HP5,HP6,
      $NOLIST,IDATSTART
       IDIM=2560 ; MODFLG=0
       IHARM=0
       IDATSTART=5
       NOLIST=1
       PI=3.141592653589793
       TWOPI=2.*PI
       DTR=PI/180.
       RTD=180./PI
       LABL(1)=4HGENR
       LABL(2)=4HANOU
       LABL(3)=4HTPUT
       LABL(4)=0
       CALL INGEN
       CALL OPAR(NERR,MF,1,1,KSUM,512)
       NBLKS=LENGTH/IDIM
       MMOD=LENGTH-IDIM*NBLKS
       NBSAVE=NBLKS
       IDIMSAVE=IDIM
C CALL NORAN
       IF (NSERIES.GT.MF) OUTPUT
      $'NO. OF SERIES CAN''T BE GREATER THAN NO. OF LOG. FILES';
      $STOP
       IF ((MF.EQ.21).AND.(NSERIES.EQ.21)) NSERIES=20 ; OUTPUT
      $'NO. OF SERIES HAS BEEN SET TO 20'
C
C
C
       DO 400 IFL=1,NSERIES
       LZ=-1
       CALL WDISC(IFL,1,LABL,4)
       IF (NBLKS.EQ.0) GO TO 375
   305 CONTINUE
C
```

```
C
      KV=IXA(IFL)
      DO 370 J=1,NBLKS
C INITIALIZE WORKING ARRAYS
C
      DO 310 I=1,IDIM
  310 B1(I)=0.
C
      IF ((KG(IFL).EQ.0).AND.(KNG(IFL).EQ.0)) GO TO 350
      IF (MOD(IDIM,2).EQ.1) GO TO 320
      IHALF=IDIM/2
C
      DO 315 K=1,IHALF
  315 CALL NORAN(1,KV,BMEAN(IFL),SDE(IFL),B1(K),B1(K+IHALF))
      GO TO 330
  320 CONTINUE
      IHALF=IDIM/2
C
      DO 325 K=1,IHALF+1
  325 CALL NORAN(1,KV,BMEAN(IFL),SDE(IFL),B1(K),B1(K+IHALF+1))
C
  330 CONTINUE
C SUBROUTINE NOISE ADDS GAUSSIAN PLUS NON-G. NOISE
      CALL NOISE
  350 CONTINUE
C
C
      IF (KHARM(IFL).EQ.0) GO TO 355
C SUBROUTINE HARMONICS ADDS 1ST THRU 6TH HARMONICS
      CALL HARMONICS
  355 CONTINUE
C
      DO 360 K=1,IDIM
      LZ=LZ+1
      DFLS=DFLOAT(LZ) ; DFLTH=DFLOAT(LENGTH)
      T=TWOPI*DFLS/DFLTH
      N=IFL
      B1(K)=B1(K)+
     $XS1(N)*DSIN(W11(N)*T+T11(N))*DSIN(W12(N)*T+T12(N))+
     $XS2(N)*DSIN(W21(N)*T+T21(N))*DSIN(W22(N)*T+T22(N))+
     $XS3(N)*DSIN(W31(N)*T+T31(N))*DSIN(W32(N)*T+T32(N))+
     $XS4(N)*DSIN(W41(N)*T+T41(N))*DSIN(W42(N)*T+T42(N))
  360 CONTINUE
C
      IF (MODFLG.EQ.1) GO TO 365
      CALL WDISC(IFL,IDATSTART+(J-1)*IDIM,B1,IDIM)
      GO TO 370
  365 CALL WDISC(IFL,IDATSTART+NBSAVE*IDIMSAVE,B1,IDIM)
  370 CONTINUE
C
C
  375 CONTINUE
      IF (MMOD.EQ.0) GO TO 390
      NBLKS=1
      MODFLG=1
      IDIM=MMOD
      MMOD=0
      GO TO 305
  390 CONTINUE
      NBLKS=NBSAVE
```

```
      IDIM=IDIMSAVE
      MODFLG=0
      MMOD=LENGTH-IDIM*NBLKS
  400 CONTINUE
C
C
C
      CALL CDISC
      STOP
C
C
C
C
      SUBROUTINE INGEN
C INTERNAL SUBROUTINE FOR PROGRAM GENRANO2 TO INITIALIZE
C AND INPUT PROGRAM VARIABLES
      IDEC=0
      WRITE (108,1000)
 1000 FORMAT ('INPUT LENGTH, NO. OF SERIES, NO. OF LOG. FILES')
      READ (105,1500) LENGTH,NSERIES,MF
 1500 FORMAT (3G)
      WRITE (108,2000)
 2000 FORMAT ('YOU WILL BE ASKED TO INPUT NAMELIST VARIABLES ',
     $'FOR EACH SERIES.',/,
     $'IG,ING,IHARM (VALUE 1 OR 0) ARE SWITCHES FOR GAUSS., ',
     $'NON-G. NOISE, AND ',/,
     $T6,'HARMONIC SERIES ** DEFAULTS 0,0,0',/,
     $'GAUSSIAN NOISE:',/,
     $'  AG,AMEAN,STDEV ARE AMPLITUDE,MEAN,ST DEV. DEFAULTS ',
     $'1,0,1',/,
     $'  "SEED" FOR SUBROUTINE NORAN IS IX. DEFAULT 314159',/,
     $'NON-GAUSSIAN NOISE:',/,
     $'  FORM IS ASQ*X**2+ACUBE*X**3 WHERE X IS GAUSS. NOISE ',
     $/,'  DEFAULTS ARE ASQ=ACUBE=1',/,
     $'HARMONIC SERIES:',/,
     $'  SIX TERMS.AMPLITUDES ARE HA1 THRU HA6. FUND FREQ. IS',
     $'  FF.',/,
     $'  PHASES HP1 THRU HP6 IN DEGREES. DEFAULTS ALL 0',/,
     $'SUM OF SIN*SIN TERMS:',/,
     $'  FOUR TERMS. AMPLITUDES ARE AS1,AS2,AS3,AS4. FREQ''S,'
     $,'PHASES (IN DEGREES)',/,
     $'  ARE FIJ,PIJ WHERE I=1,2,3,4 REFERS TO SIN*SIN TERM ',/
     $,' WITH AMPL. AS1,AS2,AS3,AS4 AND J=1,2 REFERS TO 1ST OR'
     $,' 2ND FACTOR ',/,
     $'  IN A GIVEN SIN*SIN TERM. ** DEFAULTS ALL 0.',/,
     $'INPUT KSAME=(SERIES NO.) TO DUPLICATE ALL PARAMETERS ',
     $/,'FROM A PREVIOUSLY INPUT SIGNAL.')
   50 CONTINUE
C
C
      DO 100 JJ=1,NSERIES
      WRITE (108,2500) JJ
 2500 FORMAT (/,'ENTER SIGNAL PARAMETERS FOR SERIES ',I2,' :')
      KSAME=0
      IF (IDEC.EQ.1) GO TO 60
      IG=ING=IHARM=0
      AG=ANG=1.;AS1=AS2=AS3=AS4=0.;AMEAN=0.;STDEV=1.
      ASQ=ACUBE=1.
      IX=314159
      HA1=HA2=HA3=HA4=HA5=HA6=0.
```

```
      F11=F12=F21=F22=F31=F32=P11=P12=P21=P22=P31=P32=0.
      F41=F42=P41=P42=0.
      FF=HP1=HP2=HP3=HP4=HP5=HP6=0.
      GO TO 70
   60 CONTINUE
      IX=IXA(JJ)
      IG=KG(JJ);ING=KNG(JJ);AMEAN=BMEAN(JJ);STDEV=SDE(JJ)
      IHARM=KHARM(JJ)
      AG=XG(JJ);AS1=XS1(JJ);AS2=XS2(JJ);AS3=XS3(JJ)
      AS4=XS4(JJ)
      ASQ=XNG(1,JJ);ACUBE=XNG(2,JJ)
      HA1=AH1(JJ);HA2=AH2(JJ);HA3=AH3(JJ);HA4=AH4(JJ);
      HA5=AH5(JJ);HA6=AH6(JJ);FF=FUND(JJ)
      HP1=TH1(JJ)/DTR;HP2=TH2(JJ)/DTR;HP3=TH3(JJ)/DTR
      HP4=TH4(JJ)/DTR;HP5=TH5(JJ)/DTR;HP6=TH6(JJ)/DTR
      F11=W11(JJ);F12=W12(JJ);F21=W21(JJ);F22=W22(JJ)
      F31=W31(JJ);F32=W32(JJ)
      F41=W41(JJ);F42=W42(JJ)
      P11=T11(JJ)/DTR;P12=T12(JJ)/DTR;P21=T21(JJ)/DTR
      P22=T22(JJ)/DTR;P31=T31(JJ)/DTR;P32=T32(JJ)/DTR
      P41=T41(JJ)/DTR;P42=T42(JJ)/DTR
   70 CONTINUE
      INPUT
      IF ((KSAME.NE.0).AND.(KSAME.GT.JJ))
     $OUTPUT 'KSAME TOO LARGE',KSAME,JJ ; STOP
      IF (KSAME.EQ.0) GO TO 75
      KK=KSAME
      IXA(JJ)=IXA(KK)
      KG(JJ)=KG(KK);KNG(JJ)=KNG(KK);BMEAN(JJ)=BMEAN(KK)
      KHARM(JJ)=KHARM(KK)
      SDE(JJ)=SDE(KK) ; XG(JJ)=XG(KK)
      XNG(1,JJ)=XNG(1,KK) ; XNG(2,JJ)=XNG(2,KK)
      XS1(JJ)=XS1(KK);XS2(JJ)=XS2(KK);XS3(JJ)=XS3(KK)
      XS4(JJ)=XS4(KK)
      W11(JJ)=W11(KK);W12(JJ)=W12(KK);W21(JJ)=W21(KK)
      W22(JJ)=W22(KK);W31(JJ)=W31(KK);W32(JJ)=W32(KK)
      W41(JJ)=W41(KK) ; W42(JJ)=W42(KK)
      T11(JJ)=T11(KK);T12(JJ)=T12(KK);T21(JJ)=T21(KK)
      T22(JJ)=T22(KK);T31(JJ)=T31(KK);T32(JJ)=T32(KK)
      T41(JJ)=T41(KK) ; T42(JJ)=T42(KK)
      AH1(JJ)=AH1(KK);AH2(JJ)=AH2(KK);AH3(JJ)=AH3(KK)
      AH4(JJ)=AH4(KK);AH5(JJ)=AH5(KK);AH6(JJ)=AH6(KK)
      FUND(JJ)=FUND(KK)
      TH1(JJ)=TH1(KK);TH2(JJ)=TH2(KK);TH3(JJ)=TH3(KK)
      TH4(JJ)=TH4(KK);TH5(JJ)=TH5(KK);TH6(JJ)=TH6(KK)
      GO TO 100
   75 CONTINUE
      IXA(JJ)=IX
      KG(JJ)=IG;KNG(JJ)=ING;BMEAN(JJ)=AMEAN;SDE(JJ)=STDEV
      KHARM(JJ)=IHARM
      XG(JJ)=AG;XS1(JJ)=AS1;XS2(JJ)=AS2;XS3(JJ)=AS3
      XS4(JJ)=AS4
      XNG(1,JJ)=ASQ;XNG(2,JJ)=ACUBE
      AH1(JJ)=HA1;AH2(JJ)=HA2;AH3(JJ)=HA3;AH4(JJ)=HA4;
      AH5(JJ)=HA5;AH6(JJ)=HA6;FUND(JJ)=FF
      TH1(JJ)=HP1*DTR;TH2(JJ)=HP2*DTR;TH3(JJ)=HP3*DTR
      TH4(JJ)=HP4*DTR;TH5(JJ)=HP5*DTR;TH6(JJ)=HP6*DTR
      W11(JJ)=F11;W12(JJ)=F12;W21(JJ)=F21;W22(JJ)=F22
      W31(JJ)=F31;W32(JJ)=F32
      W41(JJ)=F41 ; W42(JJ)=F42
```

```
      T11(JJ)=P11*DTR;T12(JJ)=P12*DTR;T21(JJ)=P21*DTR
      T22(JJ)=P22*DTR;T31(JJ)=P31*DTR;T32(JJ)=P32*DTR
      T41(JJ)=P41*DTR ; T42(JJ)=P42*DTR
  100 CONTINUE
C
C
      IF (NOLIST.EQ.0) GO TO 170
      WRITE (108,3000)
 3000 FORMAT (/,'PARAMETERS FOR ALL SERIES TO BE CREATED ',
     $'FOLLOW:',/)
C
C
      DO 150 JJ=1,NSERIES
      WRITE (108,3200) JJ
 3200 FORMAT (/,'S E R I E S  ',I2,' :')
      WRITE (108,3500) KG(JJ),BMEAN(JJ),SDE(JJ),IXA(JJ),
     $KNG(JJ),XNG(1,JJ),
     $XNG(2,JJ),KHARM(JJ),FUND(JJ),AH1(JJ),AH2(JJ),AH3(JJ),
     $AH4(JJ),AH5(JJ),AH6(JJ),TH1(JJ)*RTD,TH2(JJ)*RTD,
     $TH3(JJ)*RTD,TH4(JJ)*RTD,TH5(JJ)*RTD,TH6(JJ)*RTD,
     $XS1(JJ),XS2(JJ),XS3(JJ),XS4(JJ),
     $W11(JJ),T11(JJ)*RTD,W12(JJ),T12(JJ)*RTD,
     $W21(JJ),T21(JJ)*RTD,W22(JJ),T22(JJ)*RTD,
     $W31(JJ),T31(JJ)*RTD,W32(JJ),T32(JJ)*RTD,
     $W41(JJ),T41(JJ)*RTD,W42(JJ),T42(JJ)*RTD
 3500 FORMAT ('GAUSSIAN NOISE: IG=',I,' AMEAN=',G8.2,' STDEV=',
     $G8.2,' SEED=',I10,/,
     $'NON-G. NOISE: ING=',I,'  ASQ=',G8.2,' ACUBE=',G8.2,/,
     $'HARMONIC SERIES:  IHARM=',I,'   FF=',G8.2,/,
     $'HA1=',G8.2,'HA2=',G8.2,'HA3=',G8.2,'HA4=',G8.2,'HA5=',
     $G8.2,'HA6=',G8.2,/,
     $'HP1=',G8.2,'HP2=',G8.2,'HP3=',G8.2,'HP4=',G8.2,'HP5=',
     $G8.2,'HP6=',G8.2,/,
     $'SUM OF SIN*SIN TERMS:',/,' AS1=',G8.2,'   AS2=',G8.2,
     $'   AS3=',G8.2,'   AS4=',G8.2,/,
     $' F11=',G9.3,5X,'P11=',F6.1,8X,'F12=',G9.3,5X,'P12=',F6.1
     $,/,
     $' F21=',G9.3,5X,'P21=',F6.1,8X,'F22=',G9.3,5X,'P22=',F6.1
     $,/,
     $' F31=',G9.3,5X,'P31=',F6.1,8X,'F32=',G9.3,5X,'P32=',F6.1
     $,/,
     $' F41=',G9.3,5X,'P41=',F6.1,8X,'F42=',G9.3,5X,'P42=',F6.1
     $)
  150 CONTINUE
C
C
  170 CONTINUE
      WRITE (108,3700)
 3700 FORMAT ('DO YOU WISH TO RE-ENTER INPUT LOOP?',/,
     $'VARIABLES WILL REMAIN AS SET UNLESS CHANGED.',/,
     $'ANSWER YES OR NO.')
      READ (105,4000) NOYES
 4000 FORMAT (A4)
      IF ((NOYES.NE.4HYES ).AND.(NOYES.NE.4HNO  )) GO TO 170
      IF (NOYES.EQ.4HYES ) IDEC=1 ; GO TO 50
      OUTPUT 'INPUT COMPLETED'
      RETURN
      END
```

```
      SUBROUTINE HARMONICS
C FOR PROGRAM GENRAN03 TO GENERATE HARMONICS
      DOUBLE PRECISION T,PI,TWOPI,DFLS,DFLTH,FL,DFMS,DFMTH
      DOUBLE PRECISION
     $AH1(20),AH2(20),AH3(20),AH4(20),AH5(20),AH6(20),
     $TH1(20),TH2(20),TH3(20),TH4(20),TH5(20),TH6(20)
      COMMON B1(2560),KG(20),KNG(20),XG(20),XNG(2,20),FUND(20),
     $AH1,AH2,AH3,AH4,AH5,AH6,
     $TH1,TH2,TH3,TH4,TH5,TH6,
     $IDIM,IFL,TWOPI,LENGTH,MZ
C
      DO 200 K=1,IDIM
      MZ=MZ+1
      DFMS=DFLOAT(MZ) ; DFMTH=DFLOAT(LENGTH)
      T=TWOPI*DFMS/DFMTH
      N=IFL ; FL=FUND(IFL)
      B1(K)=B1(K)+
     $AH1(N)*DSIN(   FL*T+TH1(N))+AH2(N)*DSIN(2.*FL*T+TH2(N))+
     $AH3(N)*DSIN(3.*FL*T+TH3(N))+AH4(N)*DSIN(4.*FL*T+TH4(N))+
     $AH5(N)*DSIN(5.*FL*T+TH5(N))+AH6(N)*DSIN(6.*FL*T+TH6(N))
  200 CONTINUE
C
      RETURN
      END
```

```
      SUBROUTINE NOISE
C FOR PROGRAM GENRANO3 TO GENERATE GAUSSIAN AND NON-G. NOISE
      DOUBLE PRECISION T,PI,TWOPI,DFLS,DFLTH,FL,DFMS,DFMTH
      DOUBLE PRECISION
     $AH1(20),AH2(20),AH3(20),AH4(20),AH5(20),AH6(20),
     $TH1(20),TH2(20),TH3(20),TH4(20),TH5(20),TH6(20)
      COMMON B1(2560),KG(20),KNG(20),XG(20),XNG(2,20),FUND(20),
     $AH1,AH2,AH3,AH4,AH5,AH6,
     $TH1,TH2,TH3,TH4,TH5,TH6,
     $IDIM,IFL,TWOPI,LENGTH,MZ
      IF ((KG(IFL).EQ.0).OR.(KNG(IFL).EQ.0)) GO TO 250
C
      DO 200 K=1,IDIM
  200 B1(K)=XG(IFL)*B1(K)+XNG(1,IFL)*B1(K)**2+
     $XNG(2,IFL)*B1(K)**3
C
      GO TO 500
  250 CONTINUE
      IF (KG(IFL).EQ.0)  GO TO 350
C
      DO 300 K=1,IDIM
  300 B1(K)=XG(IFL)*B1(K)
C
      GO TO 500
  350 CONTINUE
C
      DO 400 K=1,IDIM
  400 B1(K)=XNG(1,IFL)*B1(K)**2+XNG(2,IFL)*B1(K)**3
C
  500 CONTINUE
      RETURN
      END
```

- PROGRAM ORDAT -

```
C              <<<<<<<  O R D A T S  >>>>>>>
C PROGRAM TO READ CONSECUTIVE FILES PRODUCED BY FRAGTAP
C AND TO CONSTRUCT LARGE RWDISC FILE
C PROGRAMMER:  G. MARTINEAU   VERSION 01   3/9/77
      DIMENSION FILE(18000),IFILE(2500),ILF(4),NAMDCB(4),
     $IBUOYVAR(4)
      EQUIVALENCE (FILE,IFILE)
      MFILE=21
      CALL DPAR(NERR,MFILE,1,1,KSUM,512)
      NWDS=18000
      READ (105,1000) NDCB,LDCB
 1000 FORMAT (2G)
C
C
      DO 600 I=1,NDCB
      READ (105,2000) IDCB,NSERIES,NSERIES,(ILF(J),J=1,NSERIES)
 2000 FORMAT (2G,NG)
C READ IN DATA
      CALL BUFFER IN(IDCB,1,FILE,NWDS,ISTAT,INWDS)
X     OUTPUT IDCB,(IFILE(K),K=1,9)
      IF (ISTAT.NE.2) OUTPUT ISTAT ; STOP 1
      IF (INWDS.NE.9) OUTPUT INWDS,IDCB ; STOP 10
      NUMVAR=IFILE(4)
      JCYC=IFILE(5)
X     OUTPUT JCYC
      IF (NUMVAR.NE.NSERIES) OUTPUT IDCB,NUMVAR,NSERIES ;
     $STOP 'DISCREPANCY IN NO. OF VARIABLES'
C
C
C
X     OUTPUT NUMVAR
      DO 150 N=1,NSERIES
  150 IBUOYVAR(N)=IFILE(N+5)
C
      DO 200 K=1,NSERIES
      LF=ILF(K)
      CALL BUFFER IN(IDCB,1,FILE(10),NWDS,ISTAT,INWDS)
      IF (INWDS.NE.JCYC) OUTPUT N,INWDS,JCYC ; STOP 50
C
      DO 170 N=1,3
  170 IFILE(N+5)=IFILE(N)
C
      IFILE(9)=IBUOYVAR(K)
X     OUTPUT LF,JCYC,INWDS
X     WRITE (108,2222) K,(IFILE(M),M=6,9),
X    $(FILE(M),M=10,24)
X2222 FORMAT (1H ,'K=',I1,2X,3A4,3X,I8,/,(5G14.4))
      CALL WDISC(LF,1,FILE(6),JCYC+4)
  200 CONTINUE
C
      WRITE (108,3000) IDCB,(IFILE(J),J=6,8),NSERIES,JCYC,
     $NSERIES,(ILF(J),J=1,NSERIES),NSERIES,
     $(IBUOYVAR(J),J=1,NSERIES)
 3000 FORMAT (/,'OUTPUT SUMMARY FOR DCB ',I2,/,T16,
     $'BUOY FILE:  ',3A4,/,T12,'NO. OF SERIES: ',I1,/,T4,
     $'DATA WDS. EACH SERIES: ',G,/,T4,
     $'WRITTEN TO LOG. FILES:  ',N(I2,2X),/,T11,
     $'BUOY VAR. NOS.: ',N(I2,2X),//)
```

```
  600 CONTINUE
C
C
C CONSTRUCT LABEL FILE
      CALL BUFFER IN(LDCB,1,IFILE,NWDS,ISTAT,INWDS)
      IF (ISTAT.NE.2) OUTPUT ISTAT ; STOP 2
      CALL WDISC(21,1,IFILE,INWDS)
      WRITE (108,4000) INWDS
 4000 FORMAT (/,1H ,I4,' WORDS WRITTEN IN LABEL L. F. NO. 21')
      CALL CDISC
      STOP 'NORMAL END'
      END
```

- PROGRAM RBPLOT05 -

```
C                       <<<<<<<  R B P L O T 0 5  >>>>>>>
C PROGRAM TO PLOT ROTARY BICOHERENCES GENERATED BY PROGRAM
C BIVEC05 AND TRANSMITTED BY CONSECUTIVE DISC FILES.
C PROGRAMMER:  GERARD H. MARTINEAU
C ORIGINATOR:  MELBOURNE G. BRISCOE
C      DATE:  OCT. 18, 1977
C*****************************************************************
      DIMENSION DATA(200,130),IBUF(1000),CONLV(15),KARD(20),
     $IDENT1(9),IDENT2(9),ICLK(4),LABL(33),CONFLV(15),
     $INAMES(18),IVAR(6),EXCH(200)
      MD1=200 ; MD2=130
      IPASS=1
      NAMELIST ISTORE1,ISTORE2,NDECPL,CMPPT,IBUOY,IFOURIER,
     $ISTORE3,ISTORE4,WIDTH
      NDECPL=IBUOY=IFOURIER=1
      ISTORE1=1 ; ISTORE2=2 ; ISTORE3=3 ; ISTORE4=4
      CMPPT=-1. ; WIDTH=8.75
      ISW=1 ; NVEC=2 ; KPASS=1
      INPUT
      READ (105,1111) KARD
 1111 FORMAT (20A4)
      WRITE (108,1122) KARD
 1122 FORMAT (X,20A4)
      DECODE (80,3333,KARD) IW1,IW2,IW3,LPIECE,SAMPSEC,FREQTIC
 3333 FORMAT (6G)
      IF ((IW1.EQ.IW2).AND.(IW1.EQ.IW3).AND.(IW2.EQ.IW3))
     $NVEC=1
      IF ((NVEC.EQ.2).AND.(IW1.NE.IW2)) ISW=3
      LPHALF=LPIECE/2 ; LPATHE=LPIECE/8
      IF (CMPPT.LT.0.) CMPPT=WIDTH*2.54/LPHALF
      LPQU=LPIECE/4 ; LP38=LPATHE*3
      LP58=LPATHE*5 ; LP34=LPATHE*6
      READ (105,1111) KARD
      WRITE (108,1122) KARD
      DECODE (80,2222,KARD) NCONLV,NCONLV,(CONLV(I),I=1,NCONLV)
     $,NCONLV,(CONFLV(I),I=1,NCONLV)
 2222 FORMAT (G,NG,NG)
      READ (105,1111) KARD
      WRITE (108,1122) KARD
      DECODE (80,4444,KARD) IDENT1,IDENT2
 4444 FORMAT (9A4,9A4)
C
C
   30 CONTINUE
C READ ROTARY BICOHERENCES INTO ARRAY DATA(200,130)
C FIRST SET ARRAY DATA TO -999.
C
C
      DO 50 J=1,MD2
C
      DO 50 I=1,MD1
   50 DATA(I,J)=-999.
C
C
      DO 60 ISQ=1,16
      CALL BIAS(IPASS)
X     OUTPUT IBIASR,IBIASC
      IF((IPASS.EQ.1).AND.(KPASS.EQ.1))ISTORED=ISTORE1;GO TO 55
```

```
      IF((IPASS.EQ.2).AND.(KPASS.EQ.1))ISTORED=ISTORE2;GO TO 55
      IF((IPASS.EQ.1).AND.(KPASS.EQ.2))ISTORED=ISTORE3;GO TO 55
      IF((IPASS.EQ.2).AND.(KPASS.EQ.2))ISTORED=ISTORE4;GO TO 55
   55 READ (ISTORED) ((DATA(IBIASR+I,IBIASC+J),I=1,LPATHE),
     $J=1,LPATHE)
C     WRITE (108,5353) ((DATA(IBIASR+I,IBIASC+J),I=1,LPATHE),
C    $J=1,LPATHE)
C5353 FORMAT (X,5G14.4)
   60 CONTINUE
C
C
C REARRANGE DATA MATRIX IF NECESSARY
      IF (KPASS.NE.2) GO TO 67
C
C
      DO 66 J=1,LPHALF/2
C
      DO 61 I=1,MD1
   61 EXCH(I)=DATA(I,J)
C
      DO 63 I=1,MD1
   63 DATA(I,J)=DATA(I,LPHALF+1-J)
C
      DO 65 I=1,MD1
   65 DATA(I,LPHALF+1-J)=EXCH(I)
C
   66 CONTINUE
C
C
   67 CONTINUE
      IF ((IPASS.EQ.2).AND.(KPASS.EQ.1)) GO TO 360
      IF ((IPASS.EQ.1).AND.(KPASS.EQ.2)) GO TO 190
      IF ((IPASS.EQ.2).AND.(KPASS.EQ.2)) GO TO 195
C READ FOURIER LABELS
C
C
   68 DO 70 I=1,6
      READ (ISTORE1) (INAMES(J),J=3*I-2,3*I),IVAR(I)
   70 CONTINUE
C
C
      READ (ISTORE1) MF
      READ (ISTORE1) LABL
      CALL PLOTS(IBUF,-1000)
      CALL PLOT(0.,0.5,-3)
      WRITE (108,1000) (CONLV(I),I=1,NCONLV)
 1000 FORMAT (/,'CONTOUR LEVELS:',/,(1H ,5G13.3,/))
      CALL DIMWH(DATA,MD1,MD2)
      CALL FLAGZ(-999.0)
      CALL NOLABL
      DEL=CMPPT/2.54
      AXLEN=LPHALF*DEL
      SAMPHR=SAMPSEC/3600.
      DFTOT=1./(2.*SAMPHR)
      DISTIC=AXLEN*FREQTIC/DFTOT
      AYLLEN=(INT(AXLEN/DISTIC)+1)*DISTIC
      AYHLEN=AXLEN/2.+0.25
      FIRSTV=-(AYLLEN/DISTIC)*FREQTIC
C PLOT IDENTIFICATION
      EXPLOT=AXLEN+AYHLEN
```

```
      WYEPLOT=AXLEN
      IF (NVEC.EQ.1) GO TO 100
      CALL SYMBOL(EXPLOT+8.3,WYEPLOT-0.8,0.315,
     $'ROTARY CROSS BICOHERENCE',-90.,24)
      GO TO 120
  100 CALL SYMBOL(EXPLOT+8.3,WYEPLOT-0.9,0.315,
     $'ROTARY AUTO BICOHERENCE',-90.,23)
  120 CONTINUE
      IF (IBUOY.EQ.0) GO TO 125
      CALL SYMBOL(EXPLOT+7.8,WYEPLOT-1.7,0.210,
     $'COMPONENTS OF VECTOR SERIES',-90.,27)
      CALL SYMBOL(EXPLOT+7.4,WYEPLOT-1.6,0.175,
     $'ORIGINAL FILE NAMES:',-90.,20)
      CALL SYMBOL(EXPLOT+7.4,WYEPLOT-5.7,0.175,
     $'VARIABLE NUMBERS:',-90.,17)
      CALL SYMBOL(EXPLOT+7.1,WYEPLOT-0.5,0.14,'FREQUENCY 1:',
     $-90.,12)
      CALL SYMBOL(EXPLOT+6.5,WYEPLOT-0.5,0.14,'FREQUENCY 2:',
     $-90.,12)
      CALL SYMBOL(EXPLOT+5.9,WYEPLOT-0.5,0.14,'FREQUENCY 3:',
     $-90.,12)
  125 CONTINUE
      IF ((MF.NE.21).OR.(IFOURIER.EQ.0)) GO TO 130
      CALL SYMBOL(EXPLOT+5.1,WYEPLOT-2.9,0.175,
     $'PROCESSING HISTORY',-90.,18)
      CALL SYMBOL(EXPLOT+4.8,WYEPLOT-0.7,0.105,LABL,-90.,72)
      CALL SYMBOL(EXPLOT+4.5,WYEPLOT-1.3,.105,LABL(19),-90.,60)
  130 CONTINUE
C
C
      IF (IBUOY.EQ.0) GO TO 150
      DO 135 I=1,3
      CALL SYMBOL(EXPLOT+7.1-0.6*(I-1),WYEPLOT-2.5,0.14,
     $INAMES(6*I-5),-90.,12)
      CALL SYMBOL(EXPLOT+6.85-0.6*(I-1),WYEPLOT-2.5,0.14,
     $INAMES(6*I-2),-90.,12)
      FVAR=FLOAT(IVAR(2*I-1))
      CALL NUMBER(EXPLOT+7.1-0.6*(I-1),WYEPLOT-7.1,0.14,FVAR,
     $-90.,-1)
      FVAR=FLOAT(IVAR(2*I))
  135 CALL NUMBER(EXPLOT+6.85-0.6*(I-1),WYEPLOT-7.1,0.14,FVAR,
     $-90.,-1)
C
C
  150 CONTINUE
      CALL SYMBOL(EXPLOT+1.15,WYEPLOT-1.0,0.210,IDENT1,-90.,36)
      CALL SYMBOL(EXPLOT+0.85,WYEPLOT-1.0,0.210,IDENT2,-90.,36)
      CALL TODAY(ICLK)
      CALL SYMBOL(EXPLOT+3.0,WYEPLOT-0.6,0.175,'TIME OF PLOT:',
     $-90.,13)
      CALL SYMBOL(EXPLOT+2.7,WYEPLOT-0.3,0.175,ICLK,-90.,16)
      CALL SYMBOL(EXPLOT+3.8,WYEPLOT-3.8,0.140,
     $'CONTOUR LEVELS AND PERCENT CONFIDENCE',-90.,37)
      BIASX=BIASY=0.
C
C
      IF (NCONLV.GT.10) NCONLV=10
      DO 185 I=1,NCONLV
      IF (I.EQ.6)                  BIASX=0. ; BIASY=BIASY-3.0
      CALL NUMBER(EXPLOT+3.3+BIASX,WYEPLOT-4.0+BIASY,0.210,
```

```
     $CONLV(I),-90.,3)
      CALL NUMBER(EXPLOT+3.3+BIASX,WYEPLOT-5.4+BIASY,0.210,
     $CONFLV(I),-90.,1)
      BIASX=BIASX-0.4
  185 CONTINUE
  190 IF (KPASS.EQ.1) GO TO 200
C ESTABLISH NEW PLOT ORIGIN
      CALL PLOT(0.,-0.25,-3)
C Y-AXIS WITH TIC MARKS ONLY, STARTING AT W1=0
      CALL AXWJS(AYLLEN,0.,' ',1,AXLEN,90.,0.,1.,DISTIC,-1,0,0,
     $0.12,.7,.0001,.14)
C Y-AXIS STARTING AT MAX W1
      CALL AXWJS(AYLLEN+AYHLEN,0.,'FREQ2 (CPH)',-11,AXLEN,90.,
     $0.,FREQTIC,DISTIC,NDECPL,0,0)
C Y-AXIS STARTING AT MIN W1
      CALL AXWJS(0.,0.,'FREQ2 (CPH)',11,AXLEN,90.,0.,FREQTIC,
     $DISTIC,NDECPL,0,0)
C X-AXIS
      FIRSTV=-(AYLLEN/DISTIC)*FREQTIC
      CALL AXDRAW(0.,AXLEN+0.2,'FREQ1 (CPH)',11,AYHLEN+AYLLEN,
     $0.,FIRSTV,FREQTIC,DISTIC,NDECPL)
C DRAW BOUNDARIES
      CALL PLOT(AYLLEN,0.,3)
      CALL PLOT(AYLLEN-AXLEN,AXLEN,2)
      CALL PLOT(AYLLEN,AXLEN,2)
      CALL PLOT(AYLLEN+AXLEN/2.,AXLEN/2.,2)
      CALL PLOT(AYLLEN,0.,2)
      CALL GRID(AYLLEN-AXLEN,AXLEN,DEL,-DEL)
      CALL WHCNTR(DATA,1,LPQU+LPHALF+1,1,LPHALF,NCONLV,CONLV)
      CALL PLOT(0.,0.,999)
  192 CONTINUE
C BEGIN PROCEDURE FOR PLOTTING NEGATIVE SUM FREQUENCIES
      IPASS=2
      GO TO 30
  195 CONTINUE
C ESTABLISH NEW PLOT ORIGIN
      CALL PLOT(0.,0.6,-3)
C Y-AXIS WITH TIC MARKS ONLY, STARTING AT W1=0
      CALL AXWJS(AYHLEN,AXLEN,' ',-1,AXLEN,-90.,0.,1.,DISTIC,
     $-1,0,0,0.12,.7,.0001,.14)
C Y-AXIS STARTING AT MAX W1
      FIRSTV=0.
      CALL AXWJS(AYHLEN+AYLLEN,AXLEN,'FREQ2 (CPH)',11,AXLEN,
     $-90.,FIRSTV,FREQTIC,DISTIC,NDECPL,0,0)
C Y-AXIS STARTING AT MIN W1
      CALL AXWJS(0.,AXLEN,'FREQ2 (CPH)',-11,AXLEN,-90.,FIRSTV,
     $FREQTIC,DISTIC,NDECPL,0,0)
C X-AXIS
      FIRSTV=-(AYHLEN/DISTIC)*FREQTIC
      CALL AXDRAW(0.,-0.2,'FREQ1 (CPH)',-11,AYHLEN+AYLLEN,
     $0.,FIRSTV,FREQTIC,DISTIC,NDECPL)
C DRAW BOUNDARIES
      CALL PLOT(AYHLEN,0.,3)
      CALL PLOT(AYHLEN-AXLEN/2.,AXLEN/2.,2)
      CALL PLOT(AYHLEN,AXLEN,2)
      CALL PLOT(AYHLEN+AXLEN,0.,2)
      CALL PLOT(AYHLEN,0.,2)
      CALL GRID(AYHLEN-AXLEN/2.,AXLEN-DEL,DEL,-DEL)
      CALL WHCNTR(DATA,1,LPQU+LPHALF+1,1,LPHALF,
     $NCONLV,CONLV)
```

```
      CALL PLOT(0.,0.,3)
      GO TO 400
C
C
  200 CONTINUE
C X-AXIS WITH TIC MARKS ONLY, STARTING AT ORIGIN
      CALL AXWJS(AXLEN,AXLEN,' ',-1,AXLEN,-90.,0.,1.,DISTIC,
     $-1,0,0,.12,.7,.0001,.14)
C X-AXIS STARTING ABOVE MAX Y VALUE
      CALL AXDRAW(AXLEN+AYHLEN,AXLEN,'FREQ1 (CPH)',11,
     $AXLEN,-90.,0.,FREQTIC,DISTIC,NDECPL)
C X-AXIS STARTING BELOW MIN Y VALUE
      CALL AXDRAW(AXLEN-AYLLEN,AXLEN,'FREQ1 (CPH)',-11,
     $AXLEN,-90.,0.,FREQTIC,DISTIC,NDECPL)
C Y-AXIS FOR TWO SERIES
      CALL AXDRAW(AXLEN-AYLLEN,AXLEN,'FREQ2 (CPH)',11,
     $2*AYLLEN,0.,FIRSTV,FREQTIC,DISTIC,NDECPL)
C DRAW BOUNDARIES
      CALL PLOT(0.,0.,3)
      CALL PLOT(AXLEN,AXLEN,2)
      CALL PLOT(1.5*AXLEN,AXLEN/2.,2)
      CALL PLOT(AXLEN,0.,2)
      CALL PLOT(0.,0.,2)
  220 CALL GRID(0.,AXLEN-DEL,DEL,-DEL)
      CALL WHCNTR(DATA,1,LPHALF+LPQU+1,1,LPHALF,NCONLV,CONLV)
  230 CALL PLOT(0.,0.,999)
  250 CONTINUE
C BEGIN PROCEDURE FOR PLOTTING NEGATIVE SUM FREQUENCIES
      IPASS=2
      GO TO 30
  360 CONTINUE
      AYHLEN=(INT(AXLEN/(2.*DISTIC))+1)*DISTIC
C ESTABLISH NEW PLOT ORIGIN
      CALL PLOT(0.,0.,-3)
C X-AXIS WITH TIC MARKS ONLY, STARTING AT W2=0
      CALL AXWJS(AYHLEN,0.,' ',1,AXLEN,90.,0.,1.,DISTIC,-1,
     $0,0,.12,.7,.0001,.14)
      YINT=AINT(AXLEN/DISTIC)
      FIRSTV=-FREQTIC*YINT
      YLTH=DISTIC*YINT
C X-AXIS STARTING AT MAX W2
      CALL AXDRAW(AYHLEN+AYLLEN,YLTH,'FREQ1 (CPH)',11,YLTH,
     $-90.,FIRSTV,FREQTIC,DISTIC,NDECPL)
C X-AXIS STARTING AT MIN W2
      CALL AXDRAW(0.,YLTH,'FREQ1 (CPH)',-11,YLTH,-90.,FIRSTV,
     $FREQTIC,DISTIC,NDECPL)
C Y-AXIS
      FIRSTV=-(AYHLEN/DISTIC)*FREQTIC
      CALL AXDRAW(0.,AXLEN+0.2,'FREQ2 (CPH)',11,AYHLEN+AYLLEN,
     $0.,FIRSTV,FREQTIC,DISTIC,NDECPL)
C DRAW BOUNDARIES
      CALL PLOT(AYHLEN,0.,3)
      CALL PLOT(AYHLEN-AXLEN/2.,AXLEN/2.,2)
      CALL PLOT(AYHLEN,AXLEN,2)
      CALL PLOT(AYHLEN+AXLEN,AXLEN,2)
      CALL PLOT(AYHLEN,0.,2)
      CALL GRID(AYHLEN-AXLEN/2.,AXLEN,DEL,-DEL)
      CALL WHCNTR(DATA,1,LPQU+LPHALF+1,1,LPHALF,NCONLV,CONLV)
  400 CALL PLOT(0.,-.25,999)
C PLOT POSITIVE SUM FREQUENCIES FOR /W2/>/W1/ IF ISW=3
```

```
      IF ((ISW.EQ.3).AND.(KPASS.EQ.1)) KPASS=2 ; IPASS=1 ;
     $ GO TO 30
      STOP 'NORMAL PROGRAM COMPLETION'
C***********************************************************************
C***********************************************************************
      SUBROUTINE BIAS(IP)
C INTERNAL SUBROUTINE FOR PROGRAM RBPLOT
    4 GO TO (6,8),IP
    6 GO TO (10,15,15,10,20,25,25,20,10,15,15,10,25,30,30,35),
     $ISQ
    8 GO TO (30,40,40,30,35,50,50,35,30,40,40,30,50,60,60,20),
     $ISQ
   10 CALL ASN(IBIASR,1)  ; GO TO 65
   15 CALL ASN(IBIASR,2)  ; GO TO 65
   20 CALL ASN(IBIASR,3)  ; GO TO 65
   25 CALL ASN(IBIASR,4)  ; GO TO 65
   30 CALL ASN(IBIASR,5)  ; GO TO 65
   35 CALL ASN(IBIASR,6)  ; GO TO 65
   40 CALL ASN(IBIASR,7)  ; GO TO 65
   50 CALL ASN(IBIASR,8)  ; GO TO 65
   60 CALL ASN(IBIASR,9)  ; GO TO 65
   65 GO TO (67,68),IP
   67 GO TO (70,70,75,75,75,75,80,80,80,80,85,85,85,85,80,85),
     $ISQ
   68 GO TO (85,85,80,80,80,80,75,75,75,75,70,70,70,70,75,70),
     $ISQ
   70 CALL ASN(IBIASC,6)  ; RETURN
   75 CALL ASN(IBIASC,5)  ; RETURN
   80 CALL ASN(IBIASC,4)  ; RETURN
   85 CALL ASN(IBIASC,2)  ; RETURN
   90 RETURN
C***********************************************************************
      SUBROUTINE ASN(IB,IS)
    5 GO TO (10,20,30,40,50,60,70,80,90),IS
   10 IB=LPHALF+1   ; RETURN
   20 IB=LP38       ; RETURN
   30 IB=LP58+1     ; RETURN
   40 IB=LPQU       ; RETURN
   50 IB=LPATHE     ; RETURN
   60 IB=0          ; RETURN
   70 IB=LPQU+1     ; RETURN
   80 IB=LP38+1     ; RETURN
   90 IB=LPHALF+1   ; RETURN
   95 RETURN
      END
```

- PROGRAM RBPLOT06 -

```
C                   <<<<<<<  R B P L O T 0 6  >>>>>>>
C PROGRAM TO PLOT ROTARY BICOHERENCES GENERATED BY PROGRAM
C BIVEC05 AND TRANSMITTED BY CONSECUTIVE DISC FILES.
C CREATES SINGLE 30 IN CALCOMP PLOT OVER ENTIRE DOMAIN
C PROGRAMMER:  GERARD H. MARTINEAU
C ORIGINATOR:  MELBOURNE G. BRISCOE
C       DATE:  NOV. 3, 1977
C**********************************************************
      DIMENSION DATA(200,130),IBUF(1000),CONLV(15),KARD(20),
     $IDENT1(9),IDENT2(9),ICLK(4),LABL(33),CONFLV(15),
     $INAMES(18),IVAR(6),EXCH(200),XDATA(130,200),IFREQ(3)
      EQUIVALENCE (DATA,XDATA)
      MD1=200 ; MD2=130
      IPASS=1
      NAMELIST ISTORE1,ISTORE2,NDECPL,CMPPT,IBUOY,IFOURIER,
     $ISTORE3,ISTORE4,ISCR,KOMPACT,WIDTH
      NDECPL=IBUOY=IFOURIER=1
      ISTORE1=1 ; ISTORE2=2 ; ISTORE3=3 ; ISTORE4=4 ; ISCR=5
      CMPPT=-1. ; WIDTH=24.
      ISW=1 ; NVEC=2 ; KPASS=1 ; KOMPACT=0
      INPUT
      READ (105,1111) KARD
 1111 FORMAT (20A4)
      WRITE (108,1122) KARD
 1122 FORMAT (X,20A4)
      DECODE (80,3333,KARD) IW1,IW2,IW3,LPIECE,SAMPSEC,FREQTIC
 3333 FORMAT (6G)
      IF (CMPPT.LT.0.) CMPPT=WIDTH*2.54/LPIECE
      IF ((IW1.EQ.IW2).AND.(IW1.EQ.IW3).AND.(IW2.EQ.IW3))
     $NVEC=1
      IF ((NVEC.EQ.2).AND.(IW1.NE.IW2)) ISW=3
C CHECK FOR EXISTENCE OF DCB ASSIGNMENTS
      CALL GETDCB(ISTORE1,LOCC)
      CALL GETDCB(ISTORE2,LOCC)
      IF (ISW.NE.3) GO TO 10
      CALL GETDCB(ISTORE3,LOCC)
      CALL GETDCB(ISTORE4,LOCC)
      CALL GETDCB(ISCR,LOCC)
   10 CONTINUE
      LPHALF=LPIECE/2 ; LPATHE=LPIECE/8
      LPQU=LPIECE/4 ; LP38=LPATHE*3
      LP58=LPATHE*5 ; LP34=LPATHE*6
      READ (105,1111) KARD
      WRITE (108,1122) KARD
      DECODE (80,2222,KARD) NCONLV,NCONLV,(CONLV(I),I=1,NCONLV)
     $,NCONLV,(CONFLV(I),I=1,NCONLV)
 2222 FORMAT (G,NG,NG)
      READ (105,1111) KARD
      WRITE (108,1122) KARD
      DECODE (80,4444,KARD) IDENT1,IDENT2
 4444 FORMAT (9A4,9A4)
C
C
   30 CONTINUE
C READ ROTARY BICOHERENCES INTO ARRAY DATA(200,130)
C FIRST SET ARRAY DATA TO -999.
C
C
```

```
      DO 50 J=1,MD2
C
      DO 50 I=1,MD1
   50 DATA(I,J)=-999.
C
C
      DO 60 ISQ=1,16
      CALL BIAS(IPASS)
X     OUTPUT IBIASR,IBIASC
      IF((IPASS.EQ.1).AND.(KPASS.EQ.1))ISTORED=ISTORE1;GO TO 55
      IF((IPASS.EQ.2).AND.(KPASS.EQ.1))ISTORED=ISTORE2;GO TO 55
      IF((IPASS.EQ.1).AND.(KPASS.EQ.2))ISTORED=ISTORE3;GO TO 55
      IF((IPASS.EQ.2).AND.(KPASS.EQ.2))ISTORED=ISTORE4;GO TO 55
   55 READ (ISTORED) ((DATA(IBIASR+I,IBIASC+J),I=1,LPATHE),
     $J=1,LPATHE)
   60 CONTINUE
C
C
C REARRANGE DATA MATRIX IF NECESSARY
      IF (KPASS.EQ.1) GO TO 80
C
C
      DO 66 J=1,LPHALF/2
C
      DO 61 I=1,MD1
   61 EXCH(I)=DATA(I,J)
C
      DO 63 I=1,MD1
   63 DATA(I,J)=DATA(I,LPHALF+1-J)
C
      DO 65 I=1,MD1
   65 DATA(I,LPHALF+1-J)=EXCH(I)
C
   66 CONTINUE
C
C
   67 CONTINUE
      IF (IPASS.EQ.2) REWIND ISCR
      WRITE (ISCR) ((DATA(I,J),I=1,LP34+1),J=LPHALF,1,-1)
C
C
      DO 75 J=1,MD2
C
      DO 75 I=1,MD1
   75 DATA(I,J)=-999.
C
C
      REWIND ISCR
      IF (IPASS.EQ.1)
     $READ (ISCR) ((XDATA(I,J),J=1,LP34+1),I=2,LPHALF+1)
      IF (IPASS.EQ.2)
     $READ (ISCR) ((XDATA(I,J),J=1,LP34+1),I=1,LPHALF)
   80 CONTINUE
      IF ((IPASS.EQ.2).AND.(KPASS.EQ.1)) GO TO 200
      IF ((IPASS.EQ.1).AND.(KPASS.EQ.2)) GO TO 300
      IF ((IPASS.EQ.2).AND.(KPASS.EQ.2)) GO TO 400
C INITIALIZING PLOT CALLS
      CALL PLOTS(IBUF,-1000)
      CALL PLOT(0.,0.,-3)
C GENERAL SETUP FOR WHCNTR
```

```
      CALL DIMWH(DATA,MD1,MD2)
      CALL FLAGZ(-999.0)
      CALL NOLABL
      DEL=CMPPT/2.54
      AXL=LPHALF*DEL
      SAMPHR=SAMPSEC/3600.
      DFTOT=1./(2.*SAMPHR)
      DISTIC=AXL*FREQTIC/DFTOT
      TRYLEN=AXL/DISTIC
      ALEN=(1+INT(TRYLEN))*DISTIC
      IF (ABS(ALEN-TRYLEN).LE.0.1) ALEN=ALEN+DISTIC
      AXLEN=2.*ALEN
      FIRSTV=-(ALEN/DISTIC)*FREQTIC
C PLOT TITLE AND LABELS
      EXPLOT=AXLEN+1.
      WYEPLOT=28.
      IF (NVEC.EQ.1) GO TO 100
      CALL SYMBOL(EXPLOT+4.3,WYEPLOT-7.9,0.420,
     $'CROSS ROTARY BICOHERENCE',-90.,24)
      GO TO 120
  100 CALL SYMBOL(EXPLOT+4.3,WYEPLOT-8.2,0.420,
     $'AUTO ROTARY BICOHERENCE',-90.,23)
  120 CONTINUE
C
C
      DO 130 I=1,6
      READ (ISTORE1) (INAMES(J),J=3*I-2,3*I),IVAR(I)
  130 CONTINUE
C
C
      READ (ISTORE1) MF
      READ (ISTORE1) LABL
      IF (IBODY.EQ.0) GO TO 150
      CALL SYMBOL(EXPLOT+3.4,WYEPLOT-1.2,0.210,
     $'COMPONENTS OF VECTOR SERIES',-90.,27)
      CALL SYMBOL(EXPLOT+3.0,WYEPLOT-1.1,0.175,
     $'ORIGINAL FILE NAMES',-90.,20)
      CALL SYMBOL(EXPLOT+3.0,WYEPLOT-5.2,0.175,
     $'VARIABLE NUMBERS',-90.,17)
      IFREQ(1)=4HFREQ ; IFREQ(2)=4HUENC
C
C
      DO 140 I=1,3
      IF (I.EQ.1) IFREQ(3)=4HY 1: ; GO TO 137
      IF (I.EQ.2) IFREQ(3)=4HY 2: ; GO TO 137
      IFREQ(3)=4HY 3:
  137 CALL SYMBOL(EXPLOT+2.7-0.6*(I-1),WYEPLOT,0.14,IFREQ,
     $-90.,12)
      CALL SYMBOL(EXPLOT+2.7-0.6*(I-1),WYEPLOT-2.0,0.14,
     $INAMES(6*I-5),-90.,12)
      CALL SYMBOL(EXPLOT+2.45-0.6*(I-1),WYEPLOT-2.0,0.14,
     $INAMES(6*I-2),-90.,12)
  140 CONTINUE
C
C
      DO 145 I=1,3
      FVAR=FLOAT(IVAR(2*I-1))
      CALL NUMBER(EXPLOT+2.7-0.6*(I-1),WYEPLOT-6.6,0.14,FVAR,
     $-90.,-1)
      CALL NUMBER(EXPLOT+2.45-0.6*(I-1),WYEPLOT-6.6,0.14,FVAR,
```

```
     $-90.,-1)
  145 CONTINUE
C
C
  150 CONTINUE
      CALL SYMBOL(EXPLOT+3.4,WYEPLOT-9.8,0.175,
     $'CONTOUR LEVELS AND PERCENT CONFIDENCE',-90.,37)
      BIASX=BIASY=0.
      IF (NCONLV.GT.10) NCONLV=10
C
C
      DO 160 I=1,NCONLV
      IF (I.EQ.6) BIASX=0. ; BIASY=BIASY-3.0
      CALL NUMBER(EXPLOT+2.9+BIASX,WYEPLOT-11.9+BIASY,0.210,
     $CONLV(I),-90.,3)
      CALL NUMBER(EXPLOT+2.9+BIASX,WYEPLOT-13.3+BIASY,0.210,
     $CONFLV(I),-90.,3)
      BIASX=BIASX-0.4
  160 CONTINUE
C
C
      CALL SYMBOL(EXPLOT+0.4,WYEPLOT-9.2,0.210,IDENT1,-90.,36)
      CALL SYMBOL(EXPLOT,WYEPLOT-9.2,0.210,IDENT2,-90.,36)
      IF ((MF.NE.21).OR.(IFOURIER.EQ.0)) GO TO 170
      CALL SYMBOL(EXPLOT+3.4,WYEPLOT-21.4,0.175,
     $'PROCESSING HISTORY',-90.,18)
      CALL SYMBOL(EXPLOT+2.9,WYEPLOT-18.8,0.175,LABL,-90.,36)
      CALL SYMBOL(EXPLOT+2.4,WYEPLOT-18.8,0.175,LABL(10),-90.,
     $48)
      CALL SYMBOL(EXPLOT+1.9,WYEPLOT-18.8,0.175,LABL(22),-90.,
     $48)
  170 CONTINUE
      CALL TODAY(ICLK)
      CALL SYMBOL(EXPLOT+1.0,WYEPLOT-22.0,0.175,
     $'TIME OF PLOT:',-90.,13)
      CALL SYMBOL(EXPLOT+0.7,WYEPLOT-21.6,0.175,ICLK,-90.,16)
C LABELLING COMPLETED. ESTABLISH NEW ORIGIN.
      CALL PLOT(0.,14.5+AXL,-3)
C BOUNDARIES OF DOMAIN, PLUS X AND Y AXES (UNLABELLED)
      CALL PLOT(AXL,-AXL,3)
      CALL PLOT(0.5*AXL,-0.5*AXL,2)
      CALL PLOT(AXL,0.,2)
      CALL PLOT(2.*AXL,0.,2)
      CALL PLOT(0.,-2.*AXL,2)
      CALL PLOT(AXL,-2.*AXL,2)
      CALL PLOT(1.5*AXL,-1.5*AXL,2)
      CALL PLOT(AXL,-AXL,2)
      OFFSET=ALEN-AXL
      CALL AXWJS(AXL,OFFSET,' ',-1,AXLEN,-90.,0.,1.,DISTIC,
     $-1,0,0,.12,.7,.0001,.14)
      CALL AXWJS(-OFFSET,-AXL,' ',1,AXLEN,0.,0.,1.,DISTIC,
     $-1,0,0,.12,.7,.0001,.14)
      IF (ISW.NE.3) GO TO 180
      CALL PLOT(2.*AXL,0.,3)
      CALL PLOT(2.*AXL,-AXL,2)
      CALL PLOT(1.5*AXL,-1.5*AXL,2)
      CALL PLOT(0.,-2.*AXL,3)
      CALL PLOT(0.,-AXL,2)
      CALL PLOT(0.5*AXL,-0.5*AXL,2)
  180 CONTINUE
```

```
C LABELLED BOUNDARIES OF PLOT
      CALL AXWJS(-OFFSET,OFFSET,'FREQ2 (CPH)',11,AXLEN,0.,
     $FIRSTV,FREQTIC,DISTIC,NDECPL,0,0,0.12,0.7,0.14,0.14)
      CALL AXWJS(2.*AXL+OFFSET,OFFSET,'FREQ1 (CPH)',11,AXLEN,
     $-90.,FIRSTV,FREQTIC,DISTIC,NDECPL,0,0,0.12,0.7,0.14,0.14)
      CALL AXWJS(-OFFSET,OFFSET,'FREQ1 (CPH)',-11,AXLEN,-90.,
     $FIRSTV,FREQTIC,DISTIC,NDECPL,0,0,0.12,0.7,0.14,0.14)
      CALL AXWJS(-OFFSET,-2.*AXL-OFFSET,'FREQ2 (CPH)',-11,
     $AXLEN,0.,FIRSTV,FREQTIC,DISTIC,NDECPL,0,0,0.12,0.7,0.14,
     $0.14)
C BEGIN CONTOURING
C POSITIVE SUM FREQUENCIES FOR /W1/ > /W2/
      CALL GRID(0.,-AXL-DEL,DEL,-DEL)
      CALL WHCNTR(DATA,1,LP34+1,1,LPHALF,NCONLV,CONLV)
      IPASS=2
      GO TO 30
  200 CONTINUE
C NEGATIVE SUM FREQUENCIES FOR /W1/ > /W2/
      CALL GRID(0.5*AXL,0.,DEL,-DEL)
      CALL WHCNTR(DATA,1,LP34+1,1,LPHALF,NCONLV,CONLV)
      IPASS=1 ; KPASS=2
      IF (ISW.NE.3) GO TO 500
      GO TO 30
  300 CONTINUE
C POSITIVE SUM FREQUENCIES FOR /W1/ < /W2/
      CALL DIMWH(XDATA,MD2,MD1)
      CALL GRID(AXL,0.,DEL,-DEL)
      CALL WHCNTR(XDATA,1,LPHALF+1,1,LP34+1,NCONLV,CONLV)
      IPASS=2
      GO TO 30
  400 CONTINUE
C NEGATIVE SUM FREQUENCIES FOR /W1/ < /W2/
      CALL GRID(0.,-0.5*AXL,DEL,-DEL)
      CALL WHCNTR(XDATA,1,LPHALF+1,1,LP34+1,NCONLV,CONLV)
  500 CONTINUE
      CALL PLOT(0.,0.,999)
      STOP 'NORMAL PROGRAM COMPLETION'
C*******************************************************************
C*******************************************************************
      SUBROUTINE BIAS(IP)
C INTERNAL SUBROUTINE FOR PROGRAM RBPLOT
    4 GO TO (6,8),IP
    6 GO TO (10,15,15,10,20,25,25,20,10,15,15,10,25,30,30,35),
     $ISQ
    8 GO TO (30,40,40,30,35,50,50,35,30,40,40,30,50,60,60,20),
     $ISQ
   10 CALL ASN(IBIASR,1)  ; GO TO 65
   15 CALL ASN(IBIASR,2)  ; GO TO 65
   20 CALL ASN(IBIASR,3)  ; GO TO 65
   25 CALL ASN(IBIASR,4)  ; GO TO 65
   30 CALL ASN(IBIASR,5)  ; GO TO 65
   35 CALL ASN(IBIASR,6)  ; GO TO 65
   40 CALL ASN(IBIASR,7)  ; GO TO 65
   50 CALL ASN(IBIASR,8)  ; GO TO 65
   60 CALL ASN(IBIASR,9)  ; GO TO 65
   65 GO TO (67,68),IP
   67 GO TO (70,70,75,75,75,75,80,80,80,80,85,85,85,85,80,85),
     $ISQ
   68 GO TO (85,85,80,80,80,80,75,75,75,75,70,70,70,70,75,70),
     $ISQ
```

```
   70 CALL ASN(IBIASC,6)   ; RETURN
   75 CALL ASN(IBIASC,5)   ; RETURN
   80 CALL ASN(IBIASC,4)   ; RETURN
   85 CALL ASN(IBIASC,2)   ; RETURN
   90 RETURN
C*************************************************************
      SUBROUTINE ASN(IB,IS)
    5 GO TO (10,20,30,40,50,60,70,80,90),IS
   10 IB=LPHALF+1     ; RETURN
   20 IB=LP38         ; RETURN
   30 IB=LP58+1       ; RETURN
   40 IB=LPQU         ; RETURN
   50 IB=LPATHE       ; RETURN
   60 IB=0            ; RETURN
   70 IB=LPQU+1       ; RETURN
   80 IB=LP38+1       ; RETURN
   90 IB=LPHALF+1     ; RETURN
   95 RETURN
      END
```

BIBLIOGRAPHIC DATA SHEET	1. Report No. WHOI-77-75	2.	3. Recipient's Accession No.
4. Title and Subtitle AUTO AND CROSS-BISPECTRAL ANALYSIS OF SCALAR AND VECTOR TIME SERIES: PROGRAMS, PROGRAM DESCRIPTIONS, AND TESTS WITH ARTIFICIAL DATA			5. Report Date April 1978
			6.
7. Author(s) Gerard H. Martineau and Melbourne G. Briscoe			8. Performing Organization Rept. No.
9. Performing Organization Name and Address Woods Hole Oceanographic Institution Woods Hole, MA 02543			10. Project/Task/Work Unit No.
			11. Contract/Grant No. OCE76-14739
12. Sponsoring Organization Name and Address National Science Foundation			13. Type of Report & Period Covered Technical
			14.

15. Supplementary Notes

16. Abstracts

This report describes a series of computer programs developed at Woods Hole Oceanographic Institution during 1977 to compute scalar bispectra and cross-bispectra (BISCAL), rotary bispectra and cross-rotary bispectra (BIVEC) and to display corresponding bicoherences (BPLOT) and rotary bicoherences (RBPLOT) as contour plots on the bi-frequency plane. The usual method of calculating the bispectrum in the frequency domain after using the Fast Fourier Transform is used here, and so there is a program for calculating the Fourier coefficients (FOURIER) as well as programs for organizing the data (FRAGTAP, ORDAT) or generating artificial series (GENRAN) if desired. There is also one special-purpose program which calculates line integrals of (scalar) bicoherences in the bifrequency plane along paths of constant $\omega_3 = \pm\omega_1 \pm \omega_2$ to determine the relative total contribution of quadratic interactions to each frequency (BISUM). After the descriptions of these programs, there is a collection of actual FORTRAN-IV listings of them arranged alphabetically.

17. Key Words and Document Analysis. 17a. Descriptors

1. Bispectra
2. Time series analysis
3. Rotary spectra

17b. Identifiers/Open-Ended Terms

17c. COSATI Field/Group

18. Availability Statement	19. Security Class (This Report) UNCLASSIFIED	21. No. of Pages 235
	20. Security Class (This Page UNCLASSIFIED	22. Price

FORM NTIS-35 (REV. 3-72) USCOMM-DC 14952-P72

Woods Hole Oceanographic Institution
WHOI-77-75

AUTO AND CROSS-BISPECTRAL ANALYSIS OF SCALAR AND VECTOR TIME SERIES: PROGRAMS, PROGRAM DESCRIPTIONS, AND TESTS WITH ARTIFICIAL DATA by Gerard H. Martineau and Melbourne G. Briscoe. 235 pages. April 1978. Prepared for the National Science Foundation under Grant OCE76-14739.

This report describes a series of computer programs developed at Woods Hole Oceanographic Institution during 1977 to compute scalar bispectra and cross-bispectra (BISCAL), rotary bispectra and cross-rotary bispectra (BIVEC) and to display corresponding bicoherences (BPLOT) and rotary bicoherences (RBPLOT) as contour plots on the bi-frequency plane. The usual method of calculating the bispectrum in the frequency domain after using the Fast Fourier Transform is used here, and so there is a program for calculating the Fourier coefficients (FOURIER) as well as programs for organizing the data (FRAGTAP, ORDAT) or generating artificial series (GENRAN) if desired. There is also one special-purpose program which calculates line integrals of (scalar) bicoherences in the bifrequency plane along paths of constant $\omega_3 = \pm\omega_1 \pm \omega_2$ to determine the relative total contribution of quadratic interactions to each frequency (BISUM). After the descriptions of these programs, there is a collection of actual FORTRAN-IV listings of them arranged alphabetically.

1. Bispectra
2. Time series analysis
3. Rotary spectra

I. Martineau, Gerard H.
II. Briscoe, Melbourne G.
III. OCE76-14739

This card is UNCLASSIFIED

Woods Hole Oceanographic Institution
WHOI-77-75

AUTO AND CROSS-BISPECTRAL ANALYSIS OF SCALAR AND VECTOR TIME SERIES: PROGRAMS, PROGRAM DESCRIPTIONS, AND TESTS WITH ARTIFICIAL DATA by Gerard H. Martineau and Melbourne G. Briscoe. 235 pages. April 1978. Prepared for the National Science Foundation under Grant OCE76-14739.

This report describes a series of computer programs developed at Woods Hole Oceanographic Institution during 1977 to compute scalar bispectra and cross-bispectra (BISCAL), rotary bispectra and cross-rotary bispectra (BIVEC) and to display corresponding bicoherences (BPLOT) and rotary bicoherences (RBPLOT) as contour plots on the bi-frequency plane. The usual method of calculating the bispectrum in the frequency domain after using the Fast Fourier Transform is used here, and so there is a program for calculating the Fourier coefficients (FOURIER) as well as programs for organizing the data (FRAGTAP, ORDAT) or generating artificial series (GENRAN) if desired. There is also one special-purpose program which calculates line integrals of (scalar) bicoherences in the bifrequency plane along paths of constant $\omega_3 = \pm\omega_1 \pm \omega_2$ to determine the relative total contribution of quadratic interactions to each frequency (BISUM). After the descriptions of these programs, there is a collection of actual FORTRAN-IV listings of them arranged alphabetically.

1. Bispectra
2. Time series analysis
3. Rotary spectra

I. Martineau, Gerard H.
II. Briscoe, Melbourne G.
III. OCE76-14739

This card is UNCLASSIFIED

Woods Hole Oceanographic Institution
WHOI-77-75

AUTO AND CROSS-BISPECTRAL ANALYSIS OF SCALAR AND VECTOR TIME SERIES: PROGRAMS, PROGRAM DESCRIPTIONS, AND TESTS WITH ARTIFICIAL DATA by Gerard H. Martineau and Melbourne G. Briscoe. 235 pages. April 1978. Prepared for the National Science Foundation under Grant OCE76-14739.

This report describes a series of computer programs developed at Woods Hole Oceanographic Institution during 1977 to compute scalar bispectra and cross-bispectra (BISCAL), rotary bispectra and cross-rotary bispectra (BIVEC) and to display corresponding bicoherences (BPLOT) and rotary bicoherences (RBPLOT) as contour plots on the bi-frequency plane. The usual method of calculating the bispectrum in the frequency domain after using the Fast Fourier Transform is used here, and so there is a program for calculating the Fourier coefficients (FOURIER) as well as programs for organizing the data (FRAGTAP, ORDAT) or generating artificial series (GENRAN) if desired. There is also one special-purpose program which calculates line integrals of (scalar) bicoherences in the bifrequency plane along paths of constant $\omega_3 = \pm\omega_1 \pm \omega_2$ to determine the relative total contribution of quadratic interactions to each frequency (BISUM). After the descriptions of these programs, there is a collection of actual FORTRAN-IV listings of them arranged alphabetically.

1. Bispectra
2. Time series analysis
3. Rotary spectra

I. Martineau, Gerard H.
II. Briscoe, Melbourne G.
III. OCE76-14739

This card is UNCLASSIFIED

Woods Hole Oceanographic Institution
WHOI-77-75

AUTO AND CROSS-BISPECTRAL ANALYSIS OF SCALAR AND VECTOR TIME SERIES: PROGRAMS, PROGRAM DESCRIPTIONS, AND TESTS WITH ARTIFICIAL DATA by Gerard H. Martineau and Melbourne G. Briscoe. 235 pages. April 1978. Prepared for the National Science Foundation under Grant OCE76-14739.

This report describes a series of computer programs developed at Woods Hole Oceanographic Institution during 1977 to compute scalar bispectra and cross-bispectra (BISCAL), rotary bispectra and cross-rotary bispectra (BIVEC) and to display corresponding bicoherences (BPLOT) and rotary bicoherences (RBPLOT) as contour plots on the bi-frequency plane. The usual method of calculating the bispectrum in the frequency domain after using the Fast Fourier Transform is used here, and so there is a program for calculating the Fourier coefficients (FOURIER) as well as programs for organizing the data (FRAGTAP, ORDAT) or generating artificial series (GENRAN) if desired. There is also one special-purpose program which calculates line integrals of (scalar) bicoherences in the bifrequency plane along paths of constant $\omega_3 = \pm\omega_1 \pm \omega_2$ to determine the relative total contribution of quadratic interactions to each frequency (BISUM). After the descriptions of these programs, there is a collection of actual FORTRAN-IV listings of them arranged alphabetically.

1. Bispectra
2. Time series analysis
3. Rotary spectra

I. Martineau, Gerard H.
II. Briscoe, Melbourne G.
III. OCE76-14739

This card is UNCLASSIFIED